NUMERICAL INVESTIGATION OF PULSED CHEMICAL VAPOR DEPOSITION OF ALUMINUM NITRIDE

Undergraduate Honors Thesis

By

Derek Endres

Department of Mechanical Engineering

The Ohio State University,

Columbus, OH 43210, USA

Advisor: Dr. Sandip Mazumder

Department of Mechanical Engineering

The Ohio State University

February 2010

Abstract

The Metal Organic Vapor Phase Epitaxy (MOVPE) of Aluminum Nitride (AlN) results not only in the growth of an AlN thin film, but also in the growth of AlN particles suspended in the gas-phase. Particle formation of AlN is unique to the MOVPE of AlN because the bond strength of AlN (11.5eV) is much larger than that of other III-V materials. This study numerically examined the effect of pulsing the precursor gases on the MOVPE of AlN as a way to curb AlN particle formation, in both horizontal and vertical reactors. Pulsing parameters such as pulse width, pulse duration, and precursor gas flow rate were varied to see the effect on growth rate and particle formation. The numerical predictions show AlN particle formation decreases significantly as the length of carrier gas pulse width increases and the deposition rate of substrate AlN can stay at approximately the same value as the steady state value with increased precursor gas flow rates. Therefore, if pulsing is introduced with relatively large carrier gas pulse width and increased precursor gas flow rates the AlN particle formation would be minimized while keeping the growth rate more or less unaffected. Numerical results also showed that pulsing has the added benefit that it increased the average growth rate (compared to steady state growth rates) because the precursors are not wasted as particles.

Acknowledgements

I would like to thank my advisor, Dr. Sandip Mazumder for helping me every step of the way in this research. Without his help and guidance none of this could have been accomplished.

My appreciation also goes to Dr. Yann Guezennec, coordinator of the Mechanical Engineering Honors Program. The overall program furthered the research experience, and the honors research classroom experience gave me the opportunity to understand my research more thoroughly.

This research was supported in part by an undergraduate honors fellowship awarded by The Ohio State College of Engineering through the Mechanical Engineering Honors Program.

ESI Group, North America, is acknowledged for providing licenses of *CFD-ACE+*™.

Finally, a special thanks to my family and friends. Without their loving support, I would not be the person I am today.

Vita

Derek Joseph Endres was born January 12, 1987 in Cincinnati, Ohio. He grew up in Loveland, Ohio with his parents Jay and Patty Endres and brothers Bruce and Shaun. After graduating from St. Xavier High School in 2005, Derek moved on to The Ohio State University. While at The Ohio State University he majored in Mechanical Engineering and minored in Computer Science Engineering. Through his engineering internships, collegiate coursework, and research efforts, Derek attained a wide breadth of knowledge that furthered the value of his degree and college experience. He is a member of the Ohio State's men's club volleyball team and a coach of a local men's high school volleyball team. Currently, Derek is studying to obtain his Masters degree in Mechanical Engineering at The Ohio State University.

Table of Contents

Abstract ii
Acknowledgements iii
Vita iv
Table of Contents v
Table of Figures vi
Table of Tables vii
1 **Introduction** 1
1.1 Technology Background 1
1.2 Motivation for Current Research 3
1.3 Objectives 6
1.4 Organization of Thesis 6
2 **Research Method** 7
2.1 Governing Equations 7
2.2 Chemical Reactions 11
2.2.1 GaN 11
2.2.2 AlN 15
2.2.3 Solution Strategy 16
3 **Results and Discussion** 25
3.1 Gallium Nitride 25
3.1.1 Horizontal Reactor 25
3.1.2 Stagnation Point (Vertical) Reactor 1 27
3.1.3 Stagnation Point (Vertical) Reactor 2 31
3.2 Steady State MOVPE of AlN 33
3.2.1 Horizontal Reactor 33
3.2.2 Thomas Swan 37
3.3 Pulsed MOVPE of AlN 37
3.3.1 Horizontal Reactor 38
3.3.2 Stagnation Point (Vertical) Reactor 47
4 **Summary and Conclusion** 50
5 **Future Work** 51
6 **References** 52
Appendix A: Sample Code 54

Table of Figures

Figure 1: MOVPE Reactor AIX200 (Aixtron AG) [2] 2
Figure 2: Schematic of a Stagnation Flow Type MOVPE Reactor, and the Processes Underlying Growth of a Thin Film by Metal Organic Vapor Phase Epitaxy (MOVPE) ... 3
Figure 3: TEM Photographs of Suspended Nanoparticles Captured During AlN MOVPE at 140 Torr and 1000 ºC [4] 4
Figure 4: Proposed Pulsed MOVPE Concept for Aluminum Nitride Thin Film Growth—the Process Whose Feasibility Will Be Studied 5
Figure 5: Illustration of How the Various Relaxation Parameters Affect Stability and Convergence [7] 9
Figure 6: Schematical Representation of the Pathways (Gas Phase and Surface) for GaN Deposition [8] 14
Figure 7: Schematic of the AlN Decomposition and Reaction Pathways [5] 15
Figure 8: CFD-GEOM Screen Shot 17
Figure 9: CFD-ACE-GUI Frontend 18
Figure 10: Typical Normalized Residual Plot 20
Figure 11: Created .Out File 22
Figure 12: Sample .CVD Output 23
Figure 13: Sample Dep File 23
Figure 14: Sample Mole File 23
Figure 15: CFD-VIEW Screen Shot 24
Figure 16: Horizontal Reactor Geometry [9] 26
Figure 17: Comparison of GaN Growth Rate: Numerical (Current Work) vs. Experimental [9] 27
Figure 18: Stagnation Point Reactor Geometry [10] 29
Figure 19: GaN Stagnation Reactor Low Carrier Gas Flow Results [10] 30
Figure 20: GaN Stagnation Reactor High Carrier Gas Flow Results [10] 30
Figure 21: Thomas Swan Reactor [8] 32
Figure 22: GaN Thomas Swan Results [8] 33
Figure 23: Comparison of Predicted Numerical Results Against Previous Work for Steady State MOVPE of AlN at Various Temperatures [9] 35
Figure 24: Effect of Pressure Variation on Steady State AlN MOVPE [9] 36
Figure 25: Sample Normalized Flow Rate Chart 40
Figure 26: AlN Deposition Rate Transient Response 41
Figure 27:AlN Particle Formation Transient Response 41
Figure 28: AlN Pulsing Results for Nominal Pulse Widths of .1 sec and Constant Precursor Gas Flow Rate in a Horizontal Reactor 44
Figure 29: AlN Pulsing Results for Nominal Pulse Widths of .1 sec and Fully Adjusted Precursor Gas Flow Rate in a Horizontal Reactor 45
Figure 30: AlN Pulsing Results for Hydrogen Pulse Widths of .2 sec and Varied Precursor Gas Flow Rate in a Horizontal Reactor 46

Table of Tables

Table 1: Solvers and Intertial Relaxations 9
Table 2: Linear Relaxation 9
Table 3: Gas Phase Reactions for GaN MOVPE [8] 12
Table 4: Surface reactions considered for GaN deposition [8] 13
Table 5: GaN Compound Names and Formulas [8] 14
Table 6: Kinetic Mechanism for the Growth of AlN [5] 16
Table 7: Boundary Conditions for MOVPE of GaN in Horizontal Reactor [9] 26
Table 8: Stagnation Point Reactor Parameters [10] 28
Table 9: GaN Thomas Swan Operating Parameters [8] 32
Table 10: AlN Operating Condition for Steady State MOVPE of AlN [9] 34
Table 11: AlN Horizontal Pressure Parameters [9] 36
Table 12: Operating Conditions for AlN MOVPE in the Thomas Swan Reactor 37
Table 13: AlN Transient Horizontal Reactor Properties 38
Table 14: Pulse Width Testing Results 39
Table 15: AlN Pulsing Results for Nominal Pulse Widths of .1 sec and Constant Precursor Gas Flow Rate in a Horizontal Reactor 44
Table 16: AlN Pulsing Results for Nominal Pulse Widths of .1 sec and Fully Adjusted Precursor Gas Flow Rate in a Horizontal Reactor 45
Table 17: AlN Pulsing Results for Hydrogen Pulse Widths of .2 sec and Varied Precursor Gas Flow Rate in a Horizontal Reactor 46
Table 18: Pulsed MOVPE AlN Results in a Horizontal Reactor 47
Table 19: Thomas Swan Transient AlN Reactor Properties 48
Table 20: Transient AlN Thomas Swan Results 48

1 Introduction

1.1 Technology Background

Aluminum Nitride (AlN) is an emerging wide band-gap III-V semiconductor material with numerous applications. A III-V semiconductor is a grown semiconductor with elements from groups III and V from the periodic table. AlN, like most III-V semiconductors, has the potential application for devices that source, detect, and control light. Due to the wide band-gap (6.2 eV) of AlN, it has the capabilities to interact with deep ultraviolet light down to 210nm [1]. Therefore, one application of AlN is a light-emitting diode (LED) due to the reduced power usage and other great benefits that LEDs offer over tradition light sources. An ultraviolet LED could be used for any application normal ultraviolet light is used. AlN also has high thermal conductivity rendering it attractive as an electronic substrate material. AlN semiconductors also have applications in high-frequency communication devices, such as mobile phones, as a surface acoustic wave sensor.

Aluminum Nitride can be grown several ways. Notable methods include chemical vapor deposition (CVD) and molecular beam epitaxy (MBE). However, CVD is the *de facto* commercial technology for growth of thin films for semiconductor applications. In the case of the growth of III-V semiconductors, such as Gallium Arsenide, Gallium Nitride, or Aluminum Nitride, a special type of CVD process known as Metal Organic Vapor Phase Epitaxy (MOVPE) is used. The name "Metal Organic" comes from the fact that the group-III precursor that is used is a metal-organic compound such as tri-methyl-gallium or tri-methyl-aluminum. The words "Vapor Phase" indicate that the precursors enter the reactor in the gas phase. The word "Epitaxy" refers to

atomic layer-by-layer growth. Epitaxy results in high-quality thin films that are appropriate for semiconductor applications. Figure 1 shows a sample horizontal reactor, though there are numerous designs of both vertical and horizontal reactors.

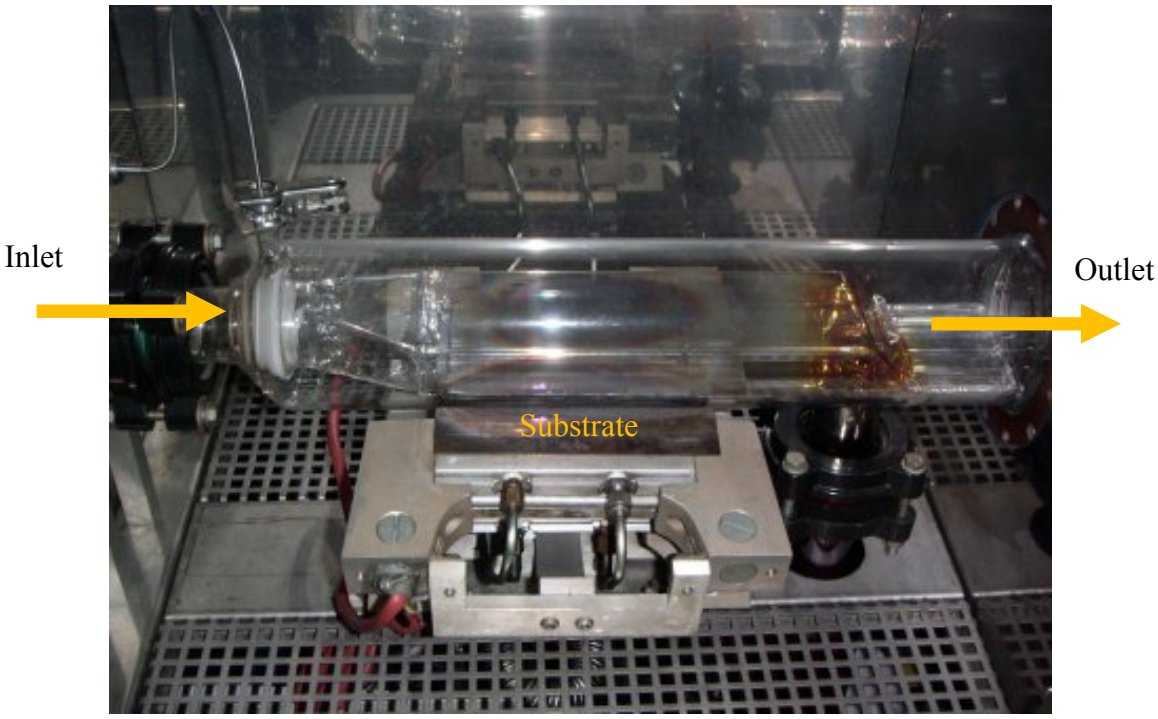

Figure 1: MOVPE Reactor AIX200 (Aixtron AG) [2]

In the MOVPE of aluminum nitride, the group-III precursor gas is tri-methyl-aluminum, $(CH_3)_3Al$ (or Teal or TMA), while the group-V precursor gas is ammonia, NH_3. These two reactants are injected into a reactor as separate streams with a carrier gas, normally hydrogen gas (H_2). Within the reactor, where the temperature is fairly high (around 700^oC is some regions), the precursor gases mix and react in the gas phase (homogeneous reactions) to produce a variety of other species, namely methane, ethane, hydrogen, nitrogen, hydrogen radicals, methyl radicals, ethyl radicals *etc*. These reactions are thermally driven reactions, often referred to as "cracking" reactions that produce smaller molecules and radicals from larger molecules. These smaller molecules and

radicals then arrive at the substrate surface. The substrate is usually a bulk crystal of the same semiconductor material. On the hot solid substrate surface, further heterogeneous (or surface) reactions occur to produce a high-quality thin film. Figure 2 depicts the entire process schematically.

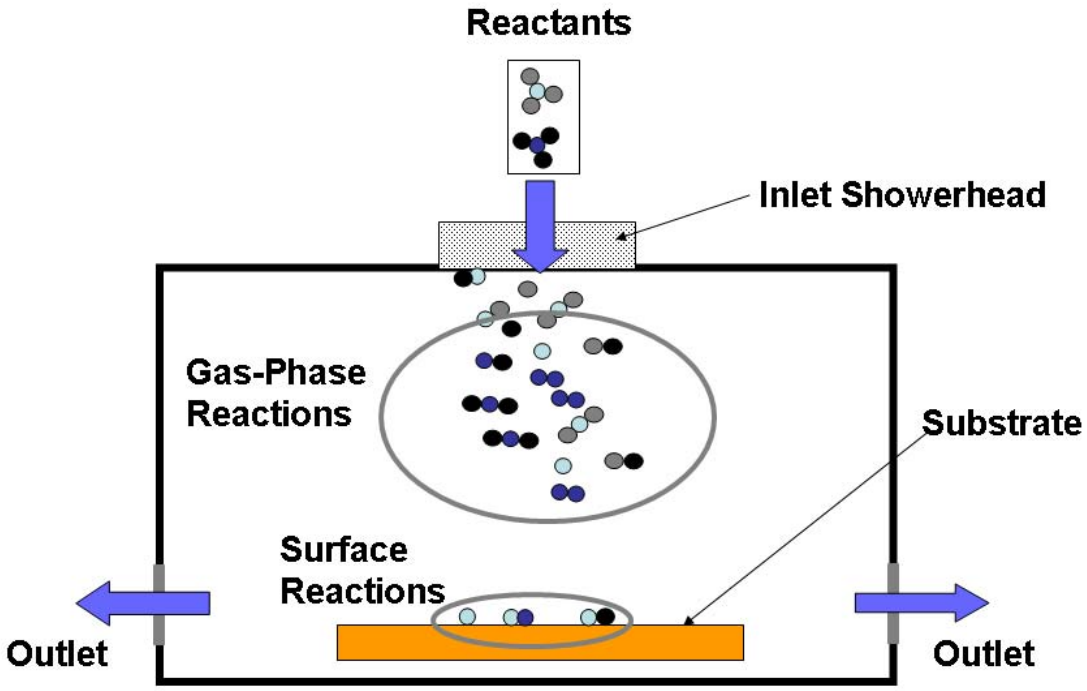

Figure 2: Schematic of a Stagnation Flow Type MOVPE Reactor, and the Processes Underlying Growth of a Thin Film by Metal Organic Vapor Phase Epitaxy (MOVPE)

There is limited experimental data for growth of AlN film. While the CVD of other III-V semiconductors have been heavily researched, aluminum nitride is not one of them, for reasons to be discussed shortly.

1.2 Motivation for Current Research

The aluminum-nitrogen system has certain unique characteristics when compared with other III-V systems. The Al-N bond strength is high compared to the bond strength of nitrogen with other group-III atoms [3]. Thus, once AlN is formed in the gas phase in

high-temperature regions of the reactor, they tend not to break apart. Furthermore, there is a strong propensity for the AlN molecules to nucleate and grow as solid suspended particles in the gas phase. This implies that much of the reactants are wasted, *i.e*. they do not contribute to the growth of the thin film [4]. Since these particles are suspended solids in the air they clog the downstream flow nozzles and pipes, resulting in damage of the reactor [5]. Furthermore, these particles often bombard the film surface, thereby deteriorating the film quality [4]. Figure 3 shows a TEM micrograph of the AlN particles that are formed during AlN MOVPE.

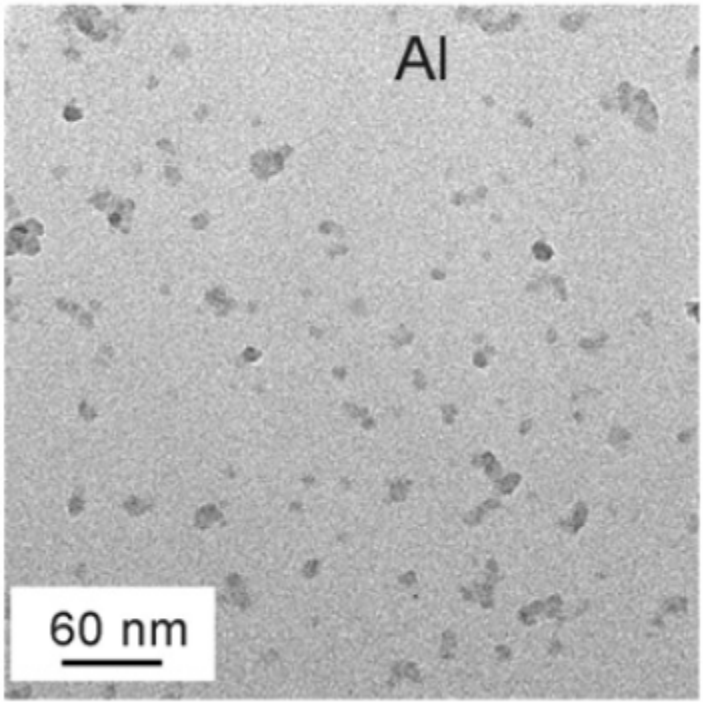

Figure 3: TEM Photographs of Suspended Nanoparticles Captured During AlN MOVPE at 140 Torr and 1000 °C [4]

It is clear from the preceding discussion, that if the gas-phase reactions can either be eliminated or suppressed, particle formation can be avoided. One way to prevent gas

phase reactions is to prevent the group-III and group-V precursors from coming into direct contact with each other. This can be accomplished by pulsing the two reactants alternately rather than injecting them continuously at all times. The carrier gas flow would be held constant. In such a scenario, the group-III precursors will arrive un-reacted to the substrate and become adsorbed. In the next pulse, the group-V reactants will be injected, and will also arrive un-reacted at the surface, where they will react with the adsorbed group-III precursors, which will result in growth of the III-V film. Figure 4 schematically shows the process.

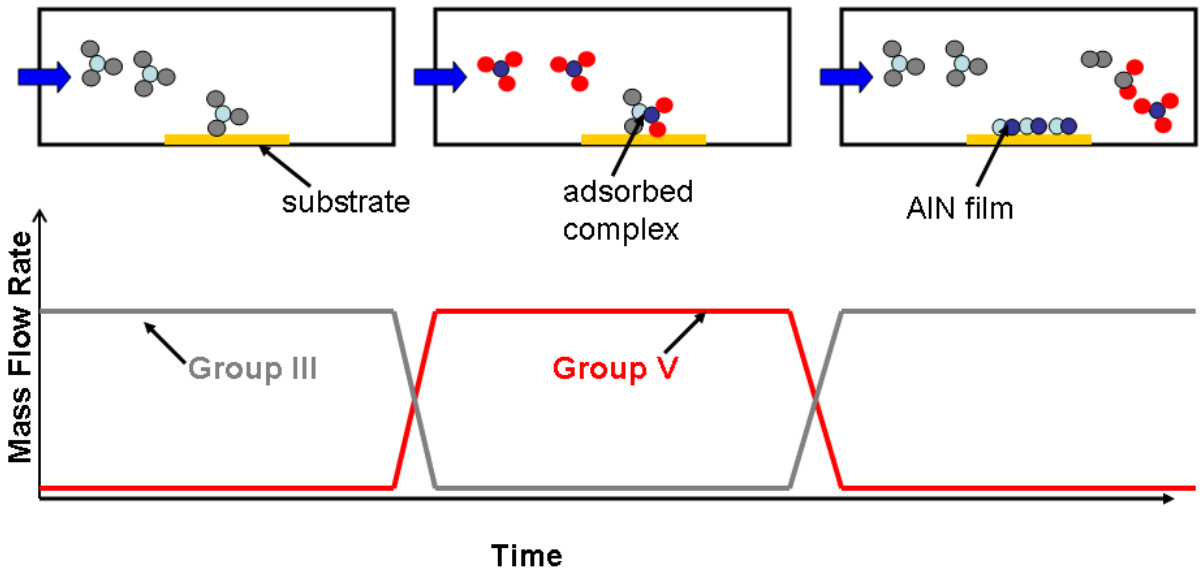

Figure 4: Proposed Pulsed MOVPE Concept for Aluminum Nitride Thin Film Growth—the Process Whose Feasibility Will Be Studied

1.3 Objectives

The objective of this research is to investigate explore if pulsed MOVPE of AlN can eliminate particle formation while retaining the same growth rate as steady state (SS) MOVPE (without pulsing) or surpassing the growth rate obtained using steady state MOVPE. First, the feasibility of pulsing the precursor gases used in the MOVPE of AlN by using detailed model-based simulations is investigated. Simulations of coupled fluid flow, heat transfer, mass transfer and chemical reactions (both gas phase and surface) will be performed to understand and explore the effect of various operating conditions and reactor geometry on film growth rate and uniformity.

Then, investigations will be conducted to understand the effect of pulse width (duration), pulse frequency and mass flow rates of the group III and V precursor on the deposition rate and AlN particle formation. Such studies will be conducted in both horizontal and vertical reactors to elucidate the various mechanisms of growth and to formulate plans for improvements of the reactor design.

1.4 Organization of Thesis

The remainder of this thesis is divided into the following chapters. Chapter 2 describes the model used to investigate AlN MOVPE, including the governing equations that were solved, the commercial software CFD-ACE™, as well as the strategy employed in arriving at the objectives. Chapter 3 shows the results of the numerical analysis and contains a discussion of the results focusing on growth rate and particle formation of AlN. Chapter 4 summarizes the results and the implications of the research results. Chapter 5 states thoughts on possible work to be done in the future.

2 Research Method

2.1 Governing Equations

The numerical analysis was performed using the general-purpose commercial fluid dynamics code, CFD-ACE™. This program solves the conservation equations of mass (Continuity Equation), momentum (Navier-Stokes Equations), energy (Enthalpy Equation) and the species conservation equations using a finite-volume approach. These equations are presented below in compact vectorial form [6].

Mass: $$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \quad (1)$$

Momentum: $$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \rho \mathbf{u}\mathbf{u} = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} \quad (2)$$

Energy: $$\frac{\partial \rho h}{\partial t} + \nabla \cdot \rho \mathbf{u} h = \nabla \cdot \mathbf{q} + \boldsymbol{\tau} : \nabla \mathbf{u} + \frac{dp}{dt} \quad (3)$$

Species: $$\frac{\partial \rho Y_i}{\partial t} + \nabla \cdot \rho \mathbf{u} Y_i = \nabla \cdot \mathbf{j}_i + \dot{\omega}_i \quad (4)$$

where ρ, $\mathbf{u}$, p, h, and Y_i are density, velocity, pressure, enthalpy and species mass-fractions, respectively. $\dot{\omega}_i$ is the species production rate per unit volume due to gas-phase reactions. The auxiliary quantities in the above equations may be written as:

Shear Stress: $$\tau = \mu\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) - \frac{2}{3}\mu(\nabla \cdot \mathbf{u})\mathbf{I} \quad (5)$$

Diffusive Energy Flux: $$\mathbf{q} = \lambda \nabla T + \sum_i h_i \mathbf{j}_i \quad (6)$$

Diffusive Species Flux: $$\mathbf{j}_i = \mathbf{j}_i^c + \mathbf{j}_i^T \quad (7)$$

Stefan-Maxwell Diffusion: $$\mathbf{j}_i^c = \rho D_i \nabla Y_i + \frac{\rho Y_i}{M} D_i \nabla M - M \sum_j D_j \nabla Y_j - \nabla M \sum_j D_j Y_j \quad (8)$$

Soret Diffusion: $$j_i^T = \frac{\rho D_i^T}{T}\nabla T - \rho Y_i \sum_j \frac{D_j^T}{T}\nabla T \quad (9)$$

where μ is the dynamic viscosity, λ is the thermal conductivity, T is the temperature, h_i are species enthalpies, D_i are species diffusivities, D_i^T are species thermal (or Soret) diffusivities, and M is the mixture molecular weight.

The physical processes occurring during MOVPE are best described by the above conservation equations. However, these equations are complex differential equations that cannot be directly solved. Their solutions are found by iterating until the residuals decrease by several orders of magnitude, typically 4. There are many possible ways to solve the equations, and some are better suited for some equations and situations. Velocity and enthalpy are solved by the CGS+PRE (conjugate gradient solver with preconditioning) method. Pressure is solved by the AMG (algebraic multi-grid) method. The species conservation equations are solved using a Coupled Solver (block Gauss-Siedel). These solvers have parameters, inertial relaxation and linear relaxation, which change the rate of convergence and can cause convergence problems if not properly tuned in. If a problem fully converges these parameters will not affect the solution, but rather, the number of iterations until convergence. Inertial relaxation is applied to the model's dependent variables. These dependent variables are solved for directly during the iterative procedure. Examples of dependent variables are velocity, pressure, enthalpy, etc. Linear relaxation is applied to all auxiliary variables. These variables are called auxiliary variables, because they are derived and computed from the solved (dependent) variables. Examples include density, pressure, temperature etc. Figure 5 shows the effect of the relaxation parameters.

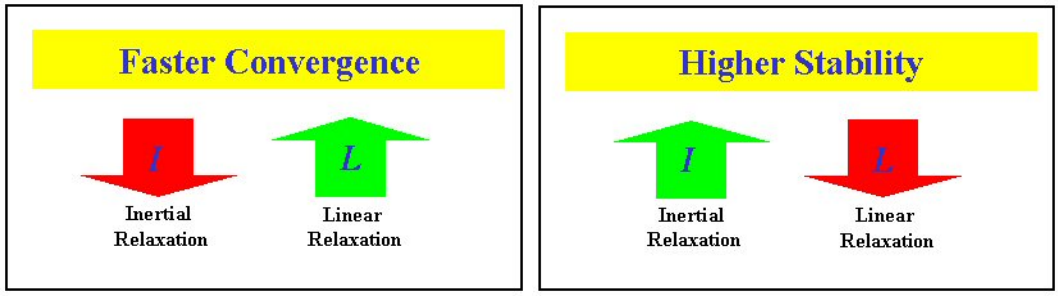

Figure 5: Illustration of How the Various Relaxation Parameters Affect Stability and Convergence [7]

Table 1 shows the nominally used solvers and inertial relaxations. Table 2 shows the nominal linear relaxation factors used.

Solver	Method	Inertial Relaxation
Velocity	CGS+PRE	.2
Pressure	AMG	.2
Enthalpy	CGS+PRE	.01
Species	Coupled Solver	.01

Table 1: Solvers and Inertial Relaxations

Variable	Linear Relaxation
Pressure	.2
Density	.2
Viscosity	.2
Temperature	.2
Species	1

Table 2: Linear Relaxation

Chemical reactions also occur, and occur in both the gas phase and at the surface. These reactions need to be modeled and this is accounted for in the code. The model for surface chemistry treatment accounts for surface-adsorbed species and surface sites. The mathematical expressions to calculate the surface growth rates are listed below. Consider a surface reaction of the general form:

$$\sum_{i=1}^{N_g} a'_{ij} A_i + \sum_{i=1}^{N_s} b'_{ij} B_i(s) + \sum_{i=1}^{N_b} c'_{ij} C_i(b) = \sum_{i=1}^{N_g} a''_{ij} A_i + \sum_{i=1}^{N_s} b''_{ij} B_i(s) + \sum_{i=1}^{N_b} c''_{ij} C_i(b) \quad (10)$$

where a_{ij}, b_{ij}, and c_{ij} are stoichiometric coefficients of gas, adsorbed, and bulk species, respectively. N_g, N_s, N_b are total numbers of gas-phase, adsorbed, and bulk (deposited) species, respectively. For this reaction, the surface reaction rate may be expressed as

$$\dot{s}_j = k_{fj} \prod_{i=1}^{N_g} [A_i]_w^{a'_{ij}} \prod_{i=1}^{N_s} [B_i(s)]^{b'_{ij}} - k_{rj} \prod_{i=1}^{N_g} [A_i]_w^{a''_{ij}} \prod_{i=1}^{N_s} [B_i(s)]^{b''_{ij}} \quad (11)$$

where k_{tj} are forward rates and k_{rj} are reverse rates, and the gas-phase concentrations at the surface are expressed as:

$$[A_i]_w = \frac{\rho_w Y_i}{M_i} \quad (12)$$

and the surface concentrations are expressed as:

$$[B_i(s)] = \rho_s X_i \quad (13)$$

with ρ_w and ρ_s being gas-phase mass density and surface site density, respectively. Y_i and X_i are the gas-phase mass-fractions and surface site fractions, respectively. A species flux balance at the reacting surface yields

$$j_i = M_i \sum_j \left(a''_{ij} - a'_{ij}\right) \dot{s}_j \qquad , i=1, \ldots, N_g \quad (14)$$

$$\frac{d[Bi(s)]}{dt} = \sum_j \left(b''_{ij} - b'_{ij}\right) \dot{s}_j \qquad , i=1, \ldots, N_s \quad (15)$$

These non-linear sets of differential algebraic equations (Eqs. (14) and (15)) are solved in a coupled Newton iteration, with the first-order backward Euler and second-order Crank-Nicolson schemes available for time integration for transient simulations.

Material properties are needed as inputs in CFD-ACE™. In some cases they are prescribed numerical values into the model. In other cases, the model uses equations to

calculate a certain material property, and is selected by the user of the program. In this specific case, the mixture density was calculated using the ideal gas law, viscosity and thermal conductivity were computed using kinetic theory, and specific heat capacity was computed using the JANNAF database. The values needed as inputs to these models were available from the database of the software. Gravity is ignored due to the small length scales involved, were resulting in very small Froude and Rayleigh numbers.

2.2 Chemical Reactions

2.2.1 GaN

Prior to exercising the code (CFD-ACE™) for pulsed MOVPE, it was validated by comparing predicted results against experimental data for GaN. GaN was chosen as the candidate material for validation studies because experimental data on GaN MOVPE is abundant, and also, reaction mechanisms for GaN growth are available in the literature.

The chemical pathway from the precursor gases to solid GaN is quite complicated. Table 3 shows the gas-phase reactions and Table 4 shows the surface reactions, responsible for GaN epitaxial growth. Table 5 list the chemical formulas involved in the chemical reactions. Figure 6 schematically shows the chemical pathways responsible for GaN growth.

	Reactions	A	n	E/R
G1	$TMG \leftrightarrow DMG + CH_3$	1.0×10^{47}	−9.18	38,750
G2	$DMG \leftrightarrow MMG + CH_3$	7.67×10^{43}	−9.8	17,120
G3	$MMG \leftrightarrow Ga + CH_3$	1.68×10^{30}	−5.07	42,290
G4	$TMG + NH_3 \rightarrow TMG \cdot NH_3$	2.28×10^{34}	−8.31	1568
G5	$TMG + NH_3 \rightarrow (CH_3)_2GaNH_2 + CH_4$	1.7×10^{4}	2.0	10,050
G7	$DMG + NH_3 \rightarrow DMG.NH_3$	4.08×10^{31}	−7.03	1628
G8	$DMG + NH_3 \rightarrow CH_3GaNH_2 + CH_4$	5.30×10^{5}	1.56	10,440
G9	$MMG + NH_3 \rightarrow MMG \cdot NH_3$	7.95×10^{24}	−5.21	1054
G10	$MMG + NH_3 \rightarrow GaNH_2 + CH_4$	8.10×10^{5}	1.3	8919
G11	$NH_3 + CH_3 \leftrightarrow NH_2 + CH_4$	3.31×10^{3}	2.51	4962
G12	$CH_3 + H_2 \leftrightarrow CH_4 + H$	1.2×10^{12}	0	6300
G13	$TMG + H \leftrightarrow DMG + CH_4$	5.0×10^{13}	0	5051
G14	$DMG + H \leftrightarrow MMG + CH_4$	5.0×10^{13}	0	5051
G15	$TMG.NH_3 \leftrightarrow MMG + 2CH_3 + NH_3$	1.33×10^{44}	−8.24	39,150
G16	$CH_3 + H + M \leftrightarrow CH_4 + M$	2.40×10^{22}	−1	0
G17	$2CH_3 \leftrightarrow C_2H_6$	2.0×10^{13}	0	0
G18	$2H + M \leftrightarrow H_2 + M$	2.0×10^{16}	0	0

Rate constants for unimolecular and bimolecular reactions are expressed as s^{-1} and $cm^3/(mol\,s)$, respectively. Backward rates are calculated from equilibrium by minimizing the Gibb's free energy.

Table 3: Gas Phase Reactions for GaN MOVPE [8]

	Reaction	A	n	E/R
S1	$MMG + S_N \rightarrow MMG(s)$	1.160×10^{5}	2.98	0
S2	$MMG(s) \rightarrow MMG + S_N$	1.115×10^{14}	0.55	5.4189×10^{4}
S3	$NH_3 + MMG(s) \rightarrow COMPM1(s)$	3.354×10^{7}	3.33	0
S4	$COMPM1(s) \rightarrow NH_3 + MMG(s)$	5.700×10^{13}	−0.16	4.1000×10^{3}
S5	$MMG + COMPM1(s) \rightarrow CH_4 + COMPM2(s)$	1.230×10^{10}	3.22	1.1800×10^{4}
S6	$NH_3 + COMPM2(s) \rightarrow COMPM3(s)$	3.354×10^{7}	3.33	0
S7	$COMPM3(s) \rightarrow NH_3 + COMPM2(s)$	5.700×10^{13}	−0.161	4.1000×10^{3}
S8	$MMG + COMPM3(s) \rightarrow CH_4 + COMPM4(s)$	1.230×10^{10}	3.22	1.1800×10^{4}
S9	$NH_3 + COMPM4(s) \rightarrow COMPM5(s)$	3.354×10^{7}	3.33	0
S10	$COMPM5(s) \rightarrow NH_3 + COMPM4(s)$	5.700×10^{13}	−0.161	4.1000×10^{3}
S11	$COMPM5(s) \rightarrow CH_4 + RINGM1(s)$	1.230×10^{7}	3.22	1.1800×10^{4}
S12	$S_G + RINGM1(s) \rightarrow RINGM2(s)$	3.354×10^{7}	3.33	0
S13	$\mathbf{RINGM2(s) \rightarrow 3H_2 + 3GaN + S_G + S_N}$	3.680×10^{9}	2.05	3.0000×10^{4}
S14	$CH_3 + Ga(s) \rightarrow MMG(s)$	1.760×10^{9}	1.39	0
S15	$MMG(s) \rightarrow CH_3 + Ga(s)$	4.540×10^{13}	0.0346	4.0000×10^{4}
S16	$NH_2 + S_G \rightarrow NH_2(s)$	3.168×10^{8}	1.83	0
S17	$GaNH_2 + S_N \rightarrow GaNH_2(s)$	2.273×10^{6}	2.247	0
S18	$GaNH_2(s) \rightarrow GaNH_2 + S_N$	4.826×10^{13}	0.614	4.2215×10^{4}
S19	$COMPM1(s) \rightarrow CH_4 + GaNH_2(s)$	1.490×10^{11}	0.609	1.3060×10^{4}
S20	$MMG + GaNH_2(s) \rightarrow COMPMM1(s)$	1.160×10^{5}	2.98	0
S21	$NH_3 + COMPMM1(s) \rightarrow COMPMM2(s)$	3.354×10^{7}	3.33	0
S22	$COMPMM2(s) \rightarrow CH_4 + COMPMM3(s)$	1.490×10^{11}	0.609	1.3060×10^{4}
S23	$MMG + COMPMM3(s) \rightarrow COMPMM4(s)$	1.160×10^{5}	2.98	0
S24	$NH_3 + COMPMM4(s) \rightarrow COMPMM5(s)$	3.345×10^{7}	3.33	0
S25	$COMPMM5(s) \rightarrow CH_4 + RINGM1(s)$	1.490×10^{11}	0.609	1.3060×10^{4}
S26	$NH_2(s) \rightarrow NH_2 + S_G$	1.450×10^{14}	0.09	3.0089×10^{4}
S27	$COMPMM1(s) \rightarrow MMG + GaNH_2(s)$	1.000×10^{14}	0.55	2.1550×10^{4}
S28	$COMPMM2(s) \rightarrow NH_3 + COMPMM1(s)$	5.700×10^{13}	−0.1	4.1000×10^{3}
S29	$COMPMM4(s) \rightarrow MMG + COMPMM3(s)$	1.000×10^{14}	0.55	2.1550×10^{4}
S30	$COMPMM5(s) \rightarrow NH_3 + COMPMM4(s)$	5.700×10^{13}	−0.1	4.1000×10^{3}
S31	$Ga + S_N \rightarrow Ga(s)$	1.000×10^{11}	1.50	0
S32	$Ga(s) + NH_2(s) \rightarrow GaNH_2(s) + S_G$	1.000×10^{25}	0.0	0
S33	$Ga(s) \rightarrow Ga + S_N$	1.000×10^{13}	0.0	2.2732×10^{4}
S34[a]	$6CH_3 + RINGM2(s) \rightarrow COM1(s)$	7.550×10^{7}	2.31	0
S35	$COM1(s) \rightarrow 6CH_3 + RINGM2(s)$	1.000×10^{13}	0.71	2.2902×10^{4}
S36	$\mathbf{COM1(s) \rightarrow 6CH_4 + 3GaN + S_G + S_N}$	4.000×10^{12}	0.0	2.5000×10^{4}
S37	$TMG + S_N \rightarrow TMG(s)$	1.160×10^{5}	2.98	0
S38	$NH_3 + TMG(s) \rightarrow TCOM1(s)$	3.354×10^{7}	3.33	0
S39	$TCOM1(s) \rightarrow CH_4 + TCOM2(s)$	1.490×10^{11}	0.609	1.6500×10^{4}
S40	$S_G + TCOM2(s) \rightarrow TCOM3(s)$	3.354×10^{7}	3.33	0
S41	$\mathbf{TCOM3(s) \rightarrow 2CH_4 + GaN + S_G + S_N}$	1.490×10^{11}	0.609	2.5000×10^{4}
S42	$TMG(s) \rightarrow TMG + S_N$	1.115×10^{14}	0.55	2.5000×10^{4}
S43	$TCOM1(s) \rightarrow NH_3 + TMG(s)$	5.700×10^{13}	−0.161	6.0000×10^{3}
S44	$TMG \cdot NH_3 + S_N \rightarrow TCOM1(s)$	1.160×10^{5}	2.98	0
S45	$TCOM1(s) \rightarrow TMG.NH_3 + S_N$	1.115×10^{14}	0.55	2.5000×10^{4}
S46	$TCOM1(s) \rightarrow 2CH_3 + MMG + NH_3 + S_N$	1.115×10^{14}	0.55	5.4189×10^{4}
S47	$MMG.NH_3 + S_N \rightarrow COMPM1(s)$	1.160×10^{5}	2.98	0
S48	$COMPM1(s) \rightarrow MMG \cdot NH_3 + S_N$	1.115×10^{14}	0.55	5.4189×10^{4}
S49	$MMG \cdot NH_3 + COMPM1(s) \rightarrow CH_4 + COMPM3(s)$	1.230×10^{10}	3.22	1.1800×10^{4}
S50	$MMG \cdot NH_3 + COMPM3(s) \rightarrow CH_4 + COMPM5(s)$	1.230×10^{10}	3.22	1.1800×10^{4}
S51	$MMG \cdot NH_3 + GaNH_2(s) \rightarrow COMPMM2(s)$	1.160×10^{5}	2.98	0
S52	$MMG \cdot NH_3 + COMPMM3(s) \rightarrow COMPMM5(s)$	1.160×10^{5}	2.98	0

The units are expressed in terms of cm, mol and s
a Concentration exponent of 1 has been used for CH_3

Table 4: Surface reactions considered for GaN deposition [8]

Compounds names	Chemical formula
COMPM1	$NH_3 \cdot MMG(s)$
COMPM2	$Ga \cdot NH_2 \cdot MMG(s)$
COMPM3	$NH_3 \cdot Ga \cdot NH_2.MMG(s)$
COMPM4	$Ga \cdot NH_2 \cdot Ga \cdot NH_2 \cdot MMG(s)$
COMPM5	$NH_3 \cdot Ga \cdot NH_2 \cdot Ga \cdot NH_2 \cdot MMG(s)$
RINGM1	$NH_2 \cdot Ga \cdot NH_2 \cdot Ga \cdot NH_2 \cdot Ga(s)$
RINGM2	$(s)NH_2 \cdot Ga \cdot NH_2 \cdot Ga \cdot NH_2 \cdot Ga(s)$
COMPMM1	$MMG \cdot GaNH_2(s)$
COMPMM2	$NH_3 \cdot MMG \cdot NH_2 \cdot Ga(s)$
COMPMM3	$NH_2 \cdot Ga \cdot NH_2 \cdot Ga(s)$
COMPMM4	$MMG \cdot NH_2 \cdot Ga \cdot NH_2 \cdot Ga(s)$
COMPMM5	$NH_3 \cdot MMG \cdot NH_2 \cdot Ga \cdot NH_2 \cdot Ga(s)$
TCOM1	$NH_3 \cdot TMG(s)$
TCOM2	$NH_2 \cdot DMG(s)$
TCOM3	$(s)NH_2 \cdot DMG(s)$
COM1	$RINGM2 \cdot CH_3$ complex

Table 5: GaN Compound Names and Formulas [8]

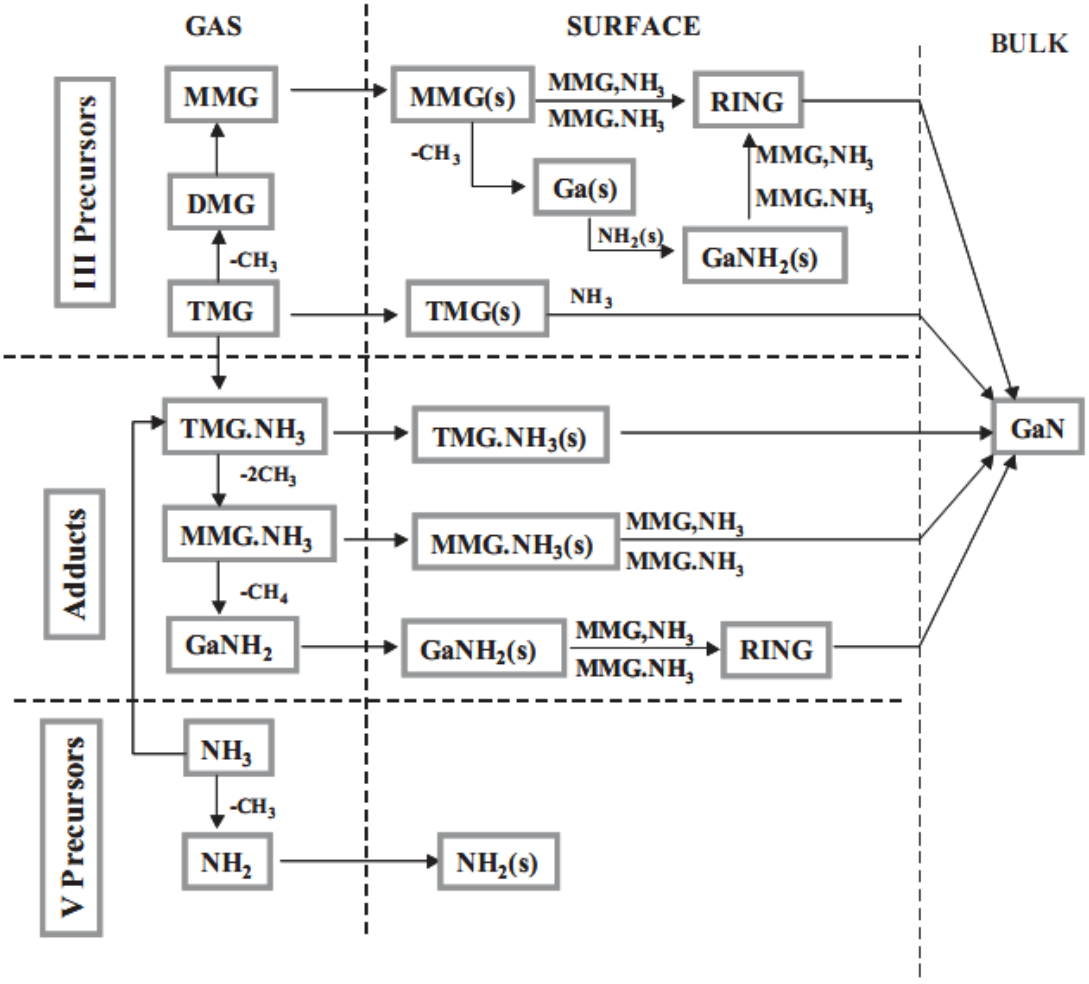

Figure 6: Schematical Representation of the Pathways (Gas Phase and Surface) for GaN Deposition [8]

2.2.2 AlN

Limited experimental data is available on AlN MOVPE due to the difficulty of growing AlN films by traditional steady state MOVPE, as discussed in Chapter 1. Moreover, there are few reported reaction mechanisms available in the literature for the growth of AlN. In this study, a reaction mechanism, proposed by Mihopoulos, Gupta, and Jensen [5] is used. Figure 7 is a schematic of the reaction pathway for AlN growth. Table 6 shows the specifics of the reactions considered in the model.

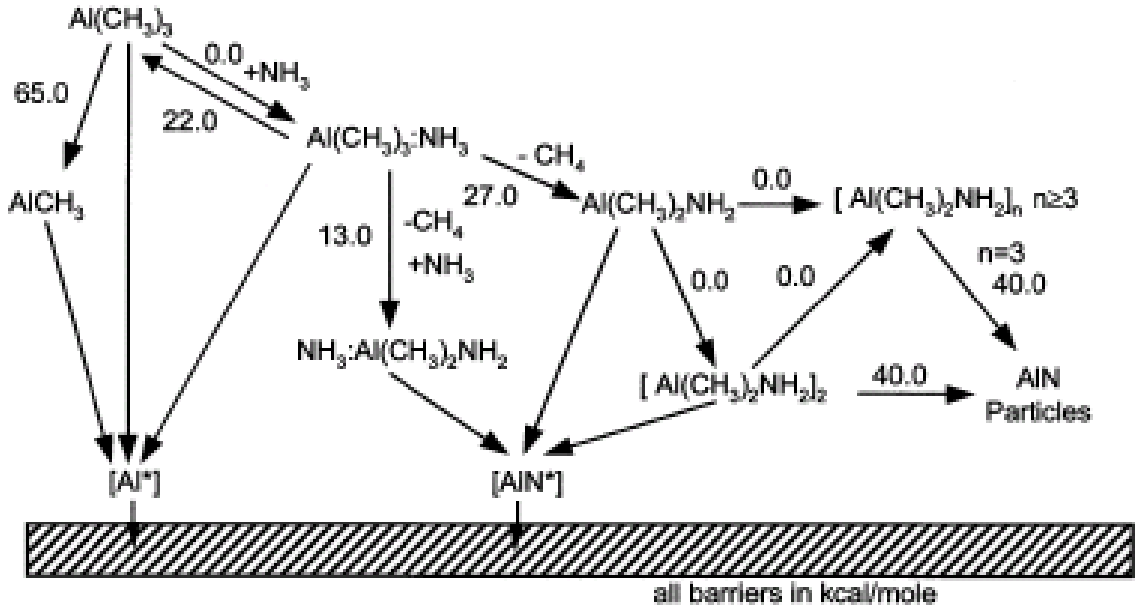

Figure 7: Schematic of the AlN Decomposition and Reaction Pathways [5]

Reaction		k	E_a
Gas phase reactions			
G1 TMAl	$\rightarrow$ MMAl + $2CH_3$	3.5×10^{15}	66.5
G2 TMAl + NH_3	$\rightarrow$ TMAl : NH_3	3.0×10^{12}	0.0
	$\leftarrow$	5.0×10^{10}	22.0
G3 TMAl : NH_3	$\rightarrow$ DMAl–NH_2 + CH_4	2.0×10^{12}	27.0
G4 TMAl : NH_3 + NH_3	$\rightarrow$ DMAl–NH_2 + CH_4 + NH_3	2.0×10^{12}	13.0
G5 2DMAl–NH_2	$\rightarrow$ $(DMAl–NH_2)_2$	4.0×10^{11}	0.0
G6 DMAl–NH_2 + $(DMAl–NH_2)_2$	$\rightarrow$ $(DMAl–NH_2)_3$	1.0×10^{11}	0.0
G7 DMAl–NH_2 + $(DMAl–NH_2)_n$	$\rightarrow$ $(DMAl–NH_2)_{n+1}$	1.0×10^{10}	0.0
G8 $(DMAl–NH_2)_2$ + $(DMAl–NH_2)_n$	$\rightarrow$ $(DMAl–NH_2)_{n+2}$	1.0×10^{10}	0.0
G9 $(DMAl–NH_2)_2$	$\rightarrow$ AlN (particle)	1.0×10^{11}	40.0
G10 $(DMAl–NH_2)_3$	$\rightarrow$ AlN (particle)	1.0×10^{11}	40.0
Surface reactions			
S1 TMAl + s	$\rightarrow$ Al* + $3CH_3$	*coll* ($\sigma = 0.1$)	0.0
S2 TMAl : NH_3 + s	$\rightarrow$ Al* + $3CH_3$ + NH_3	*coll* ($\sigma = 0.1$)	0.0
S3 MMAl + s	$\rightarrow$ Al* + CH_3	*coll* ($\sigma = 1.0$)	0.0
S4 DMAl–NH_2 + s	$\rightarrow$ AlN* + $2CH_4$	*coll* ($\sigma = 1.0$)	0.0
S5 $(DMAl–NH_2)_2$ + s	$\rightarrow$ 2 AlN* + $4CH_4$	*coll* ($\sigma = 1.0$)	0.0
S6 Al*	$\rightarrow$ AlN(s) + s	6.0×10^6	20.0
S7 AlN*	$\rightarrow$ AlN(s) + s	6.0×10^6	20.0

Note: Activation energies are in kcal/mol and pre-exponentials are in $(cm^3/mol)^{n-1}\,s^{-1}$ (for gas-phase reactions) where n is the order of the reaction and in $(cm^2/mol)\,s^{-1}$ for surface reactions.
σ denotes the sticking coefficient for the collisional type surface reaction.
Acronyms used above are: MMAl = $AlCH_3$, TMAl = $Al(CH_3)_3$, TMAl : NH_3 = $Al(CH_3)_3 : NH_3$, DMAl–NH_2 = $(CH_3)_2Al–NH_2$.

Table 6: Kinetic Mechanism for the Growth of AlN [5]

2.2.3 Solution Strategy

Modeling the MOVPE of AlN and GaN required four major steps: creation of the reactor geometry and mesh generation, setting up the boundary condition and material properties, as well as solution parameters (convergence criteria, iterations, relaxation factors, etc…), solving the governing equations, and finally, post processing. Since each case studied required different model setups, including different model geometries, the specific features of each model will be discussed in Chapter 3. The geometry file and mesh were created using the program CFD-GEOM. Figure 8 is a screen shot of the CFD-GEOM program showing a sample reactor geometry. The reactor construction in CFD-GEOM varied because some reactor models took advantage of symmetry. All the models were 2-D models of 3-D reactor using either an implied depth or symmetry. These approximations were done in the effort to increase the speed of simulation. The program

was also used to develop a mesh capable of sufficiently resolving the physics at all length scales. Mesh dependency tests were performed to make sure that the results were unaffected by mesh size for each geometry used. The horizontal reactor mesh uses a total of 4609 nodes, which corresponded to 4375 cells. The first vertical reactor looked at mesh uses 3473 nodes corresponding to 3180 cells. The second vertical, Thomas Swan, reactor looked at mesh uses 1232 nodes corresponding to 1065 cells. Figure 8 shows a sample mesh that was generated using CFD-GEOM.

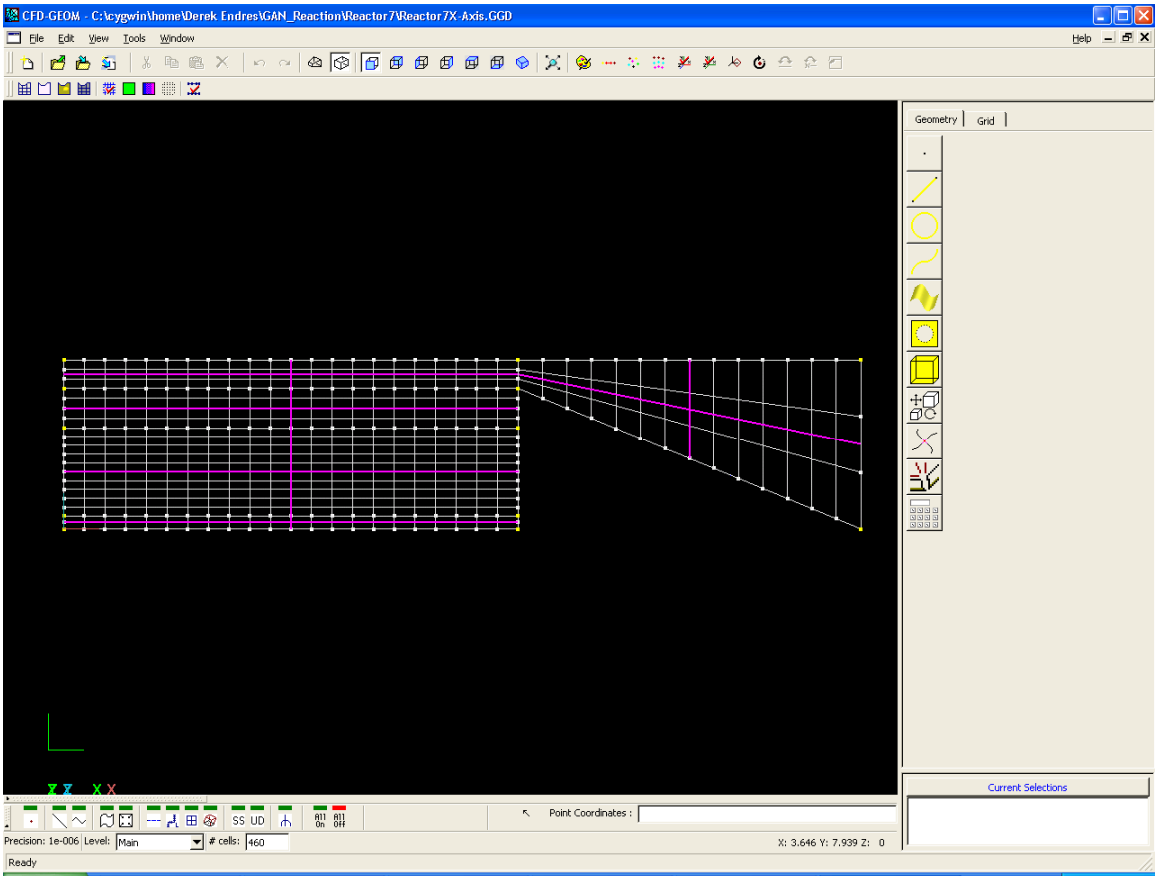

Figure 8: CFD-GEOM Screen Shot

This mesh was generated by the automatic mesh creator. This mesh file was saved and the data was transferred into another program called CFD-ACE-GUI, via a

proprietary file format, known as a .DTF file format. Figure 9 shows the display portion of the CFD-ACE-GUI program.

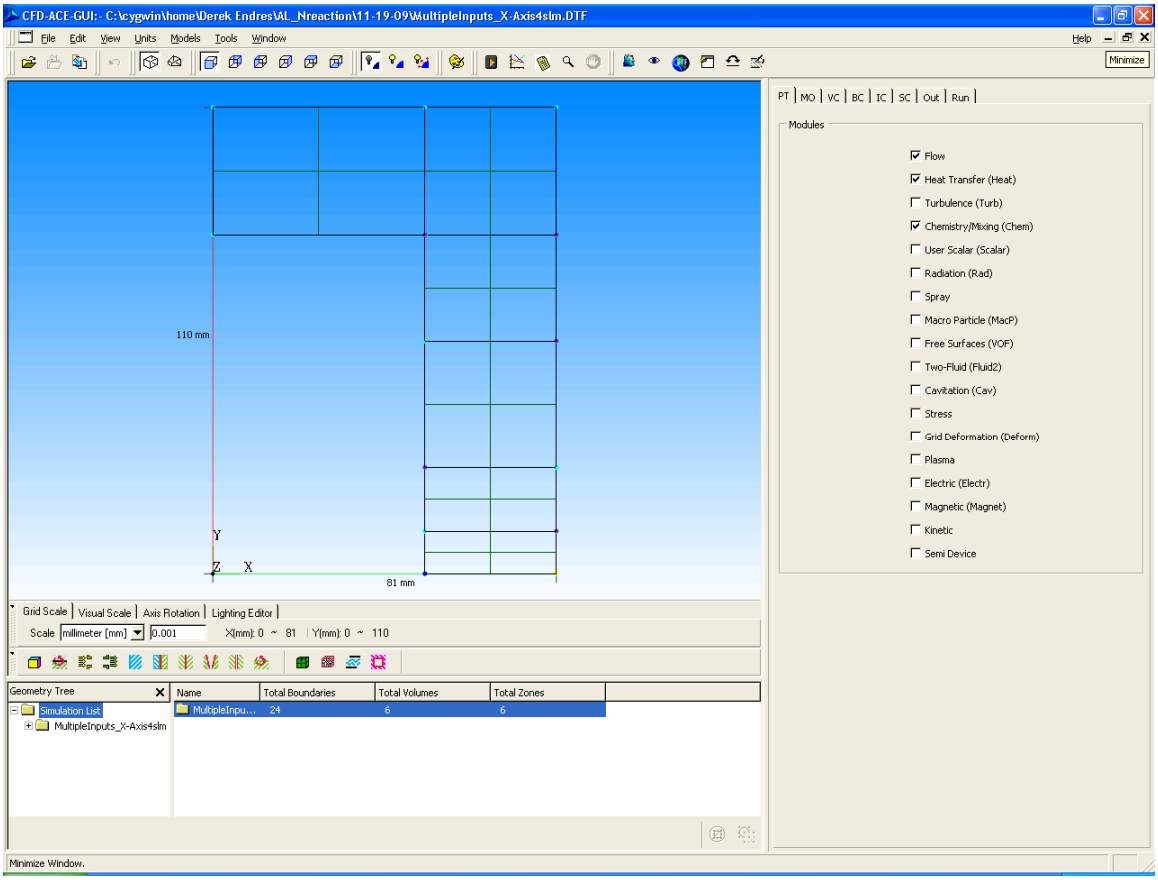

Figure 9: CFD-ACE-GUI Frontend

The purpose of CFD-ACE-GUI is to set up the boundary conditions and material properties for the simulation at hand. Some boundary conditions that are set are temperature, inlet flow rates, reactor pressure, and the reactive and inactive section of the inner wall, among others. Diffusion was turned off at the inlet, to ensure that the fluid did not flow back into the inlet. For transient (unsteady) cases, user-defined boundary conditions had to be implemented, because the input boundary conditions changed based on time. To implement the changing boundary conditions, a user-defined program had to be written in Fortran95, and the program had to use certain functions provided by the

software vendor as a way to interact with the core CFD-ACE™ solver. The Fortran95 file had to be compiled by linking it with CFD-ACE™ files into a dynamic link library (.DLL file format) that the program could interface with. This was done by using the CFD-ACE+ User Subroutine Remote Compilation online, which required loading the Fortran95 file and downloading the .DLL file. This .DLL had to be put in the same folder as the .DTF file and the correct options in the CFD-ACE-GUI had to be checked to tell the program when to use this subroutine and what it was called. This user-written program had two main parts the "uBound" section and the "uOut" section, which will discussed later, and is based upon a format given by the software vendor. A sample program is shown in the appendix A.1. The "uBound" section had to check to see where in the simulation it was run and when it was called and then set the boundary condition to the appropriate values. Once the simulation was correctly setup, the CFD-ACE™ solver (which solves the equations described in 2.1) determined the steady state or transient solution.

The solver employs an iterative technique to attain convergence based upon solution techniques chosen by the user. The residuals of each of the governing equations are tracked as they progress towards the desired convergence level. Figure 10 shows a typical normalized residual plot.

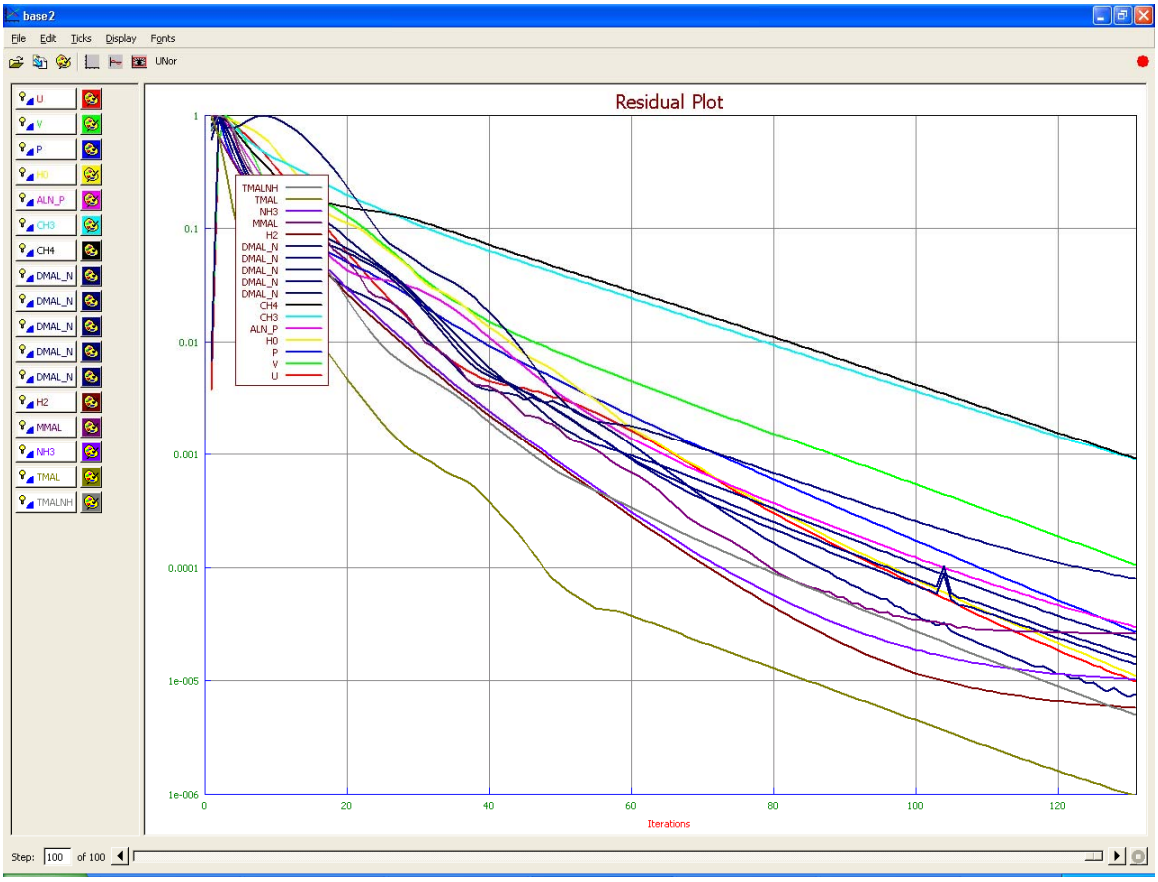

Figure 10: Typical Normalized Residual Plot

For the pulsed MOVPE studies the simulations start with a complete purge of the carrier gas. This is followed by pulses of the precursor gases and carrier gas in a repeated pulse cycle. The growth rate of the film and all other dependent variables fluctuate over time. The pulsed model is run until when the pulse cycle is repeated and a quasi steady state or periodic steady state is reached. All relevant data is then time-averaged over a cycle to get an "average" value, which is then compared to the steady state value.

CFD-ACE™ generated all the outputs that are discussed in the results, some are created by user-defined routines and some are standard outputs of the program. Distributions of all relevant quantities such as flow velocity, temperature, chemical concentration, etc., are saved in the original .DTF file. Beyond saving the relevant

quantities, the program has options to output the data in a separate file. Figure 11 shows a small portion of the ASCII output file that CFD-ACE™ prints with relevant processed data and all input parameters, known as the .Out file. Figure 12 shows the other standard ASCII output file that CFD-ACE™ prints, known as the .CVD file, which has the growth rate at each node of the wafer. The standard output files are appended at each time step. A separate .DTF file can either be saved for each time step, or the original .DTF can be overwritten. The output generated by the user-subroutine included the growth rate on the front edge of the wafer, an average of the growth rate over the entire length of the reactor, the amount of TMAL, H_2, and NH_3 entering the reactor, and the amount of AlN particles leaving the reactor for each time step. The "uOut" section of the user-code created two comma delimited text (.txt file format) files for ease of reading and interpreting. The two files hold deposition and molar flow rate information and are titled "Dep" and "Mole" files respectively. Figure 13 shows the user-defined "Dep" file. Figure 14 shows the user-defined "Mole" file.

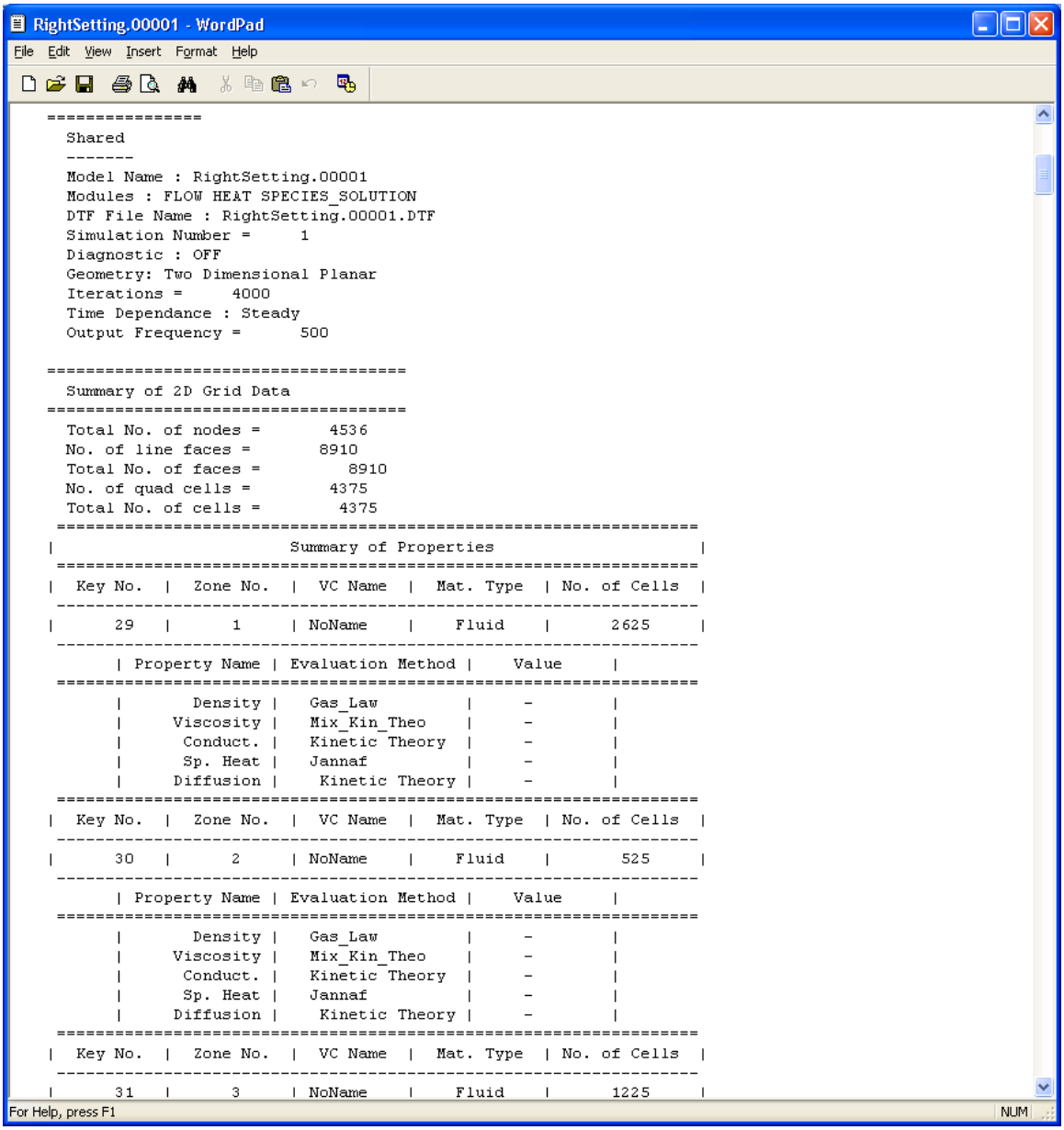

Figure 11: Created .Out File

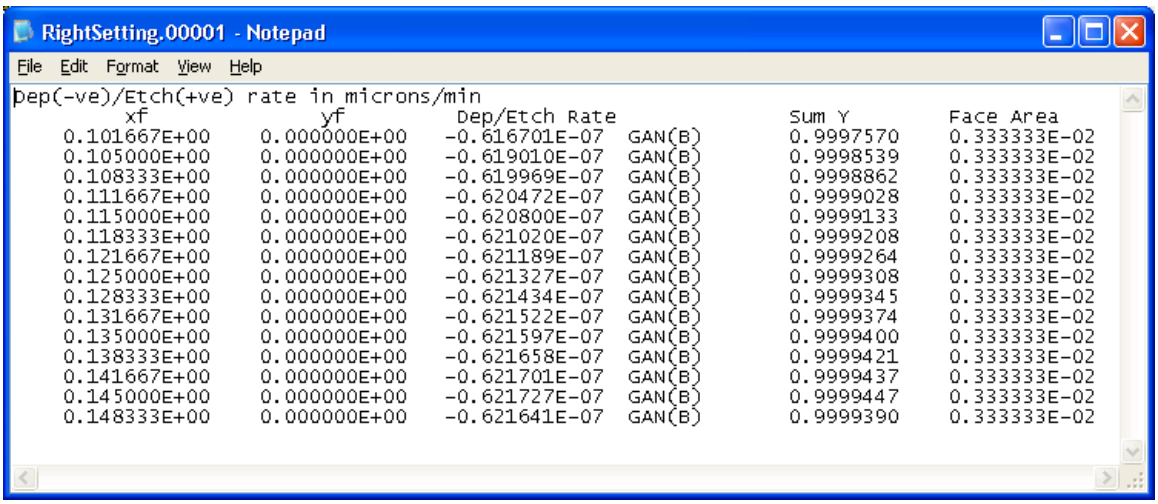

Figure 12: Sample .CVD Output

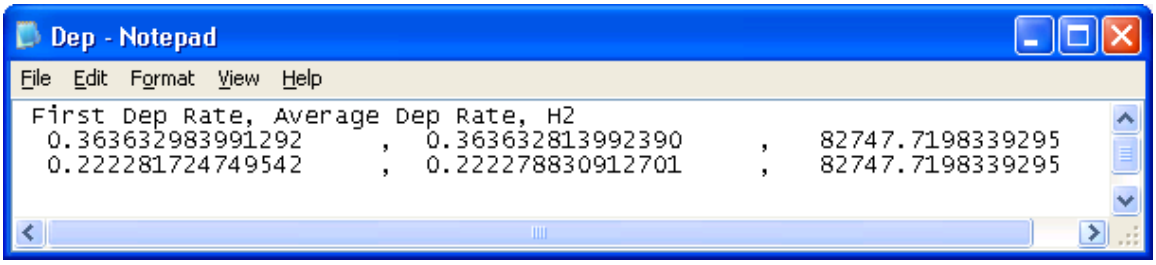

Figure 13: Sample Dep File

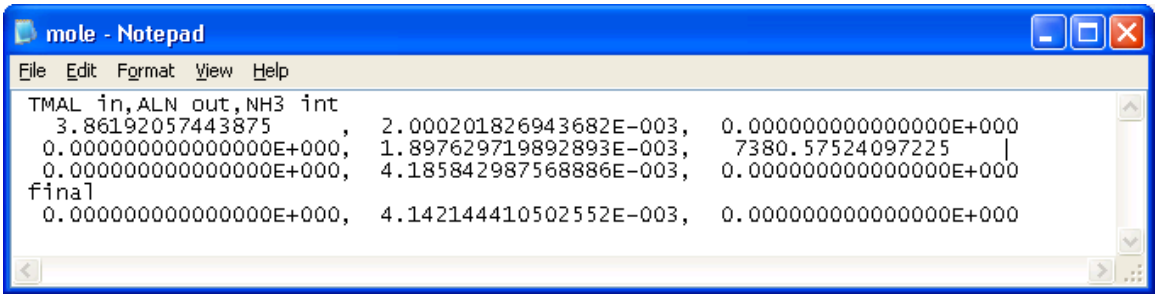

Figure 14: Sample Mole File

In order to graphically view and analyze the data, the .DTF file can be opened by CFD-VIEW, the post-processing program. Figure 15 shows a snapshot of CFD-VIEW showing AlN particle concentration at steady state. The separate .DTF files of a single simulation from the transient cases can also be linked together in CFD-VIEW to generate an animation video of the entire sequence of events.

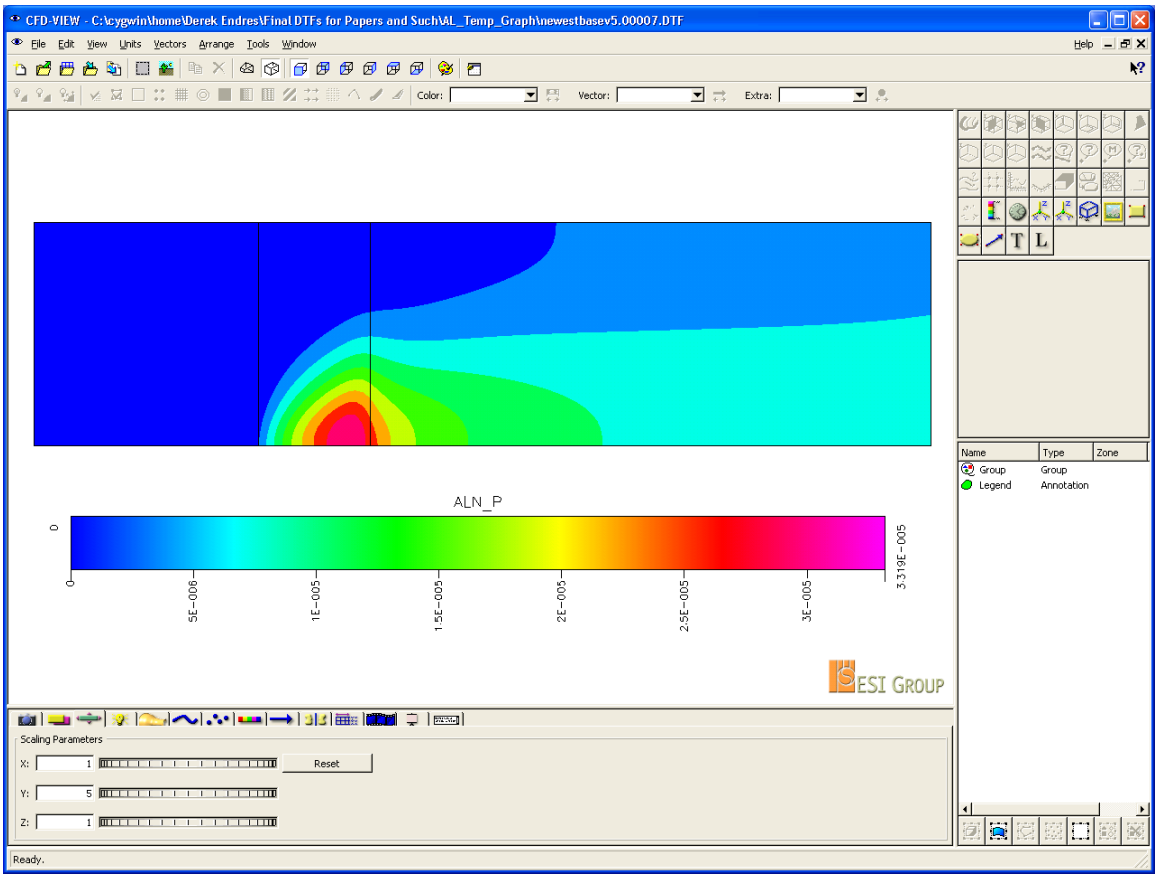

Figure 15: CFD-VIEW Screen Shot

3 Results and Discussion

The MOVPE of aluminum nitride has the undesirable result of the formation of AlN particles suspended in gas. Particle formation of AlN is unique to the MOVPE of AlN because the bond strength of AlN (11.5eV) is much larger than that of other III-V materials, such as InN (7.7eV) and GaN (8.9eV). As a result, once formed in the gas-phase, the Al-N bond tends to stay intact. Because of this difficulty, there is not much experimental data and not many models on the MOVPE of AlN. In contrast, a lot of experimental data is available for GaN MOVPE because there is no suspended particle formation. In order to validate the model used for the MOVPE of AlN, the models were first validated for GaN. The models were first validated using steady state data and then un-steady tests were conducted. The models used were tested in both horizontal and vertical reactors to ensure the results are not reactor dependent.

3.1 Gallium Nitride

3.1.1 Horizontal Reactor

The growth rate of Gallium Nitride in the horizontal reactor is highly affected by both reactor pressure and wafer temperature. A simple horizontal reactor is a good way to check the model validity by varying wafer temperature to see if the predicted numerical results match the experimental results. Figure 16 shows the horizontal reactor chosen for this study. Relevant boundary conditions are summarized in Table 7. As noted in Table 7 the temperature of the wafer is varied from 700 K to 1300 K.

Boundary Condition	Value	Units
Reactor Pressure	85	torr
TMGA* Flow Rate	30	µmol/min
H_2 Flow Rate	5000	sccm
NH_3 Flow Rate	1000	sccm
Inlet Temperature	300	ºK
Inactive Wall Temperature	300	ºK
Wafer Temperature	700 to 1300	ºK

Table 7: Boundary Conditions for MOVPE of GaN in Horizontal Reactor [9]

* TMGA - tri-methyl-gallium - $(CH_3)_3Ga$

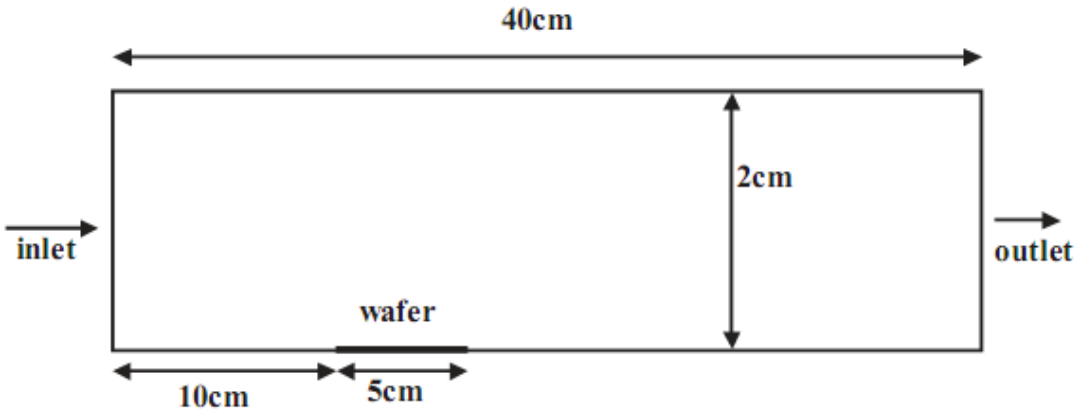

Figure 16: Horizontal Reactor Geometry [9]

Figure 17 shows a comparison of the predicted numerical results with experimental data.

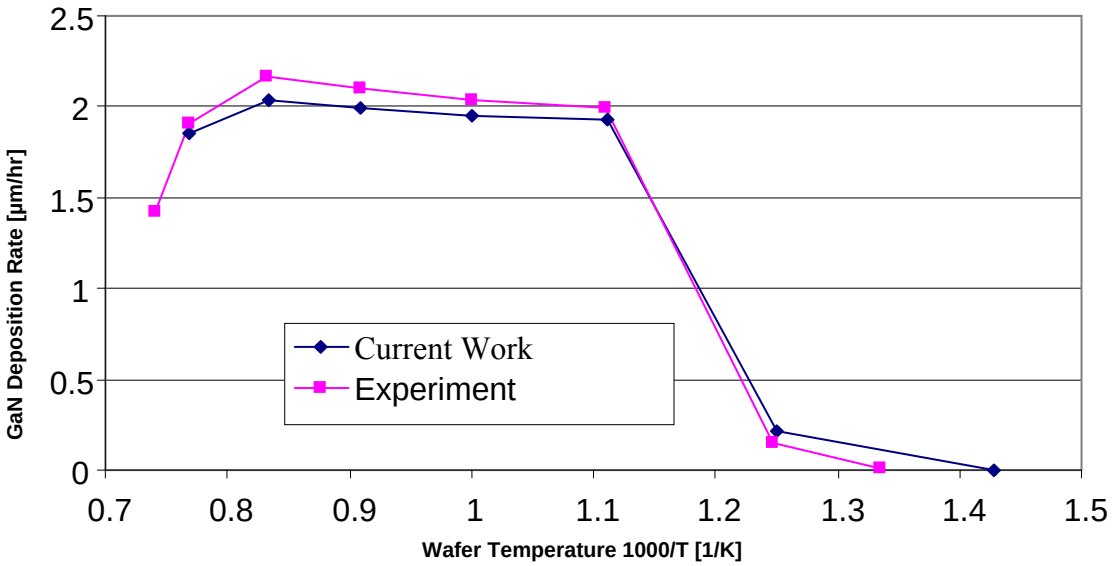

Figure 17: Comparison of GaN Growth Rate: Numerical (Current Work) vs. Experimental [9]

These results agree very well. The reason for the low growth rates at low temperatures is that the temperature is not high enough to get over the activation temperature of the chemical reactions, the so-called the "kinetically-limited regime". The reason for the low growth rates at high temperatures is that the temperature is too high and the activation barriers of the desorption reactions are crossed causing absorbed species to desorb from the surface. The middle region is the most important region. This is where the growth rate is at a maximum and is called the diffusion-limited regime. This simply means that at this temperature range, the surface reactions occur very quickly and the process is limited by the diffusion of the species to the surface.

3.1.2 Stagnation Point (Vertical) Reactor 1

A vertical reactor is much different than a horizontal reactor because the incoming fluid flows directly toward the wafer, causing a stagnation point on the wafer surface. There is also a distinct possibility of recirculation patterns developing in the corners of

the reactor. This causes the fluid flow to behave in a much more complex manner. Vertical reactor tests are included in the test procedure to ensure that the model is accurate in all situations, and not just for horizontal reactors. Figure 18 shows the vertical reactor chosen for this study and relevant boundary conditions are summarized in Table 8. For this reactor geometry, only one parameter was changed, namely the flow rate of the carrier gas.

Boundary Condition	**Value**	**Units**
Reactor Pressure	100	torr
TMGA Flow Rate	.248	sccm
H_2 Flow Rate	.2 or 10	slpm
NH_3 Flow Rate	2	slpm
Inlet Temperature	300	ºK
Inactive Wall Temperature	300	ºK
Wafer Temperature	1273	ºK

Table 8: Stagnation Point Reactor Parameters [10]

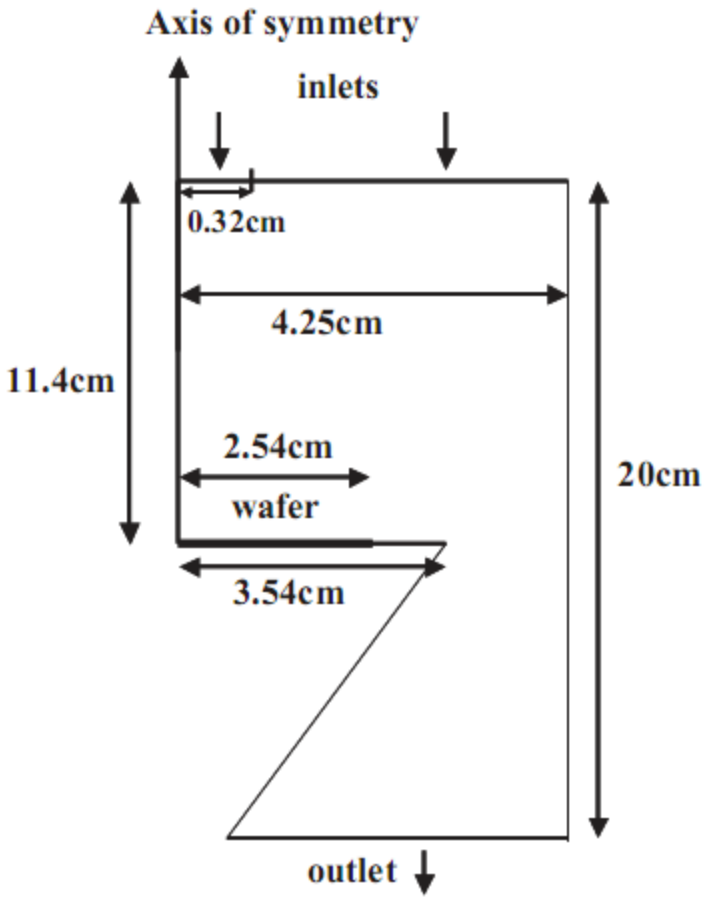

Figure 18: Stagnation Point Reactor Geometry [10]

In this study it is assumed that the TMGA and half of the total flow of the carrier gas enters through the small inlet, while NH_3 and the remaining half of the carrier gas enters through the larger inlet. Figure 19 shows a comparison of the numerical and experimental results for the low carrier gas flow rate. Figure 20 shows the comparison for the high flow rates of the carrier gas.

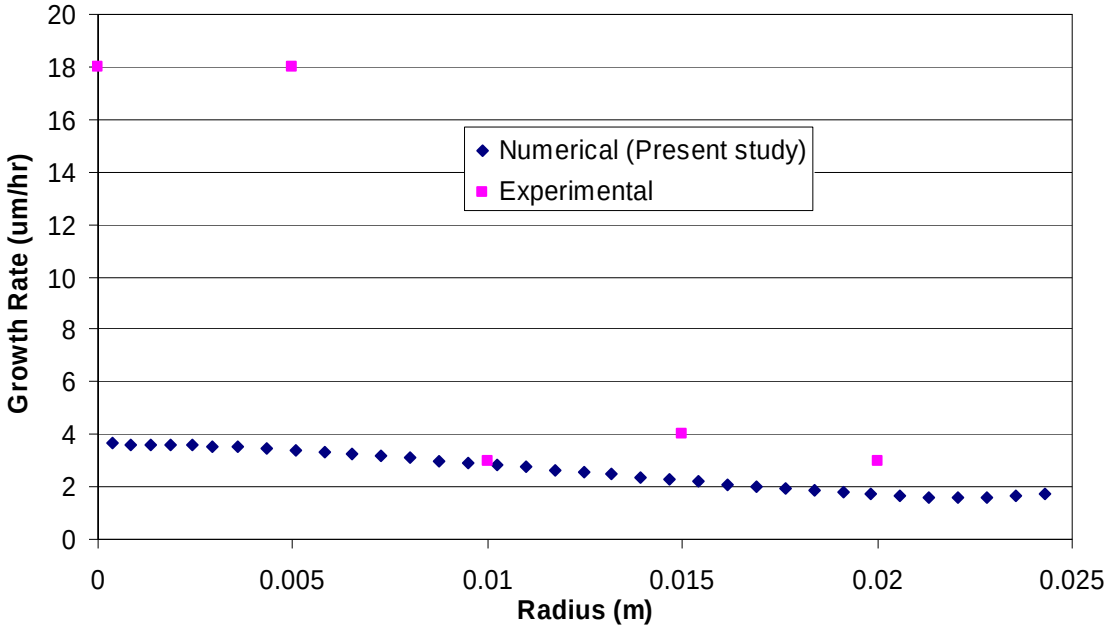

Figure 19: GaN Stagnation Reactor Low Carrier Gas Flow Results [10]

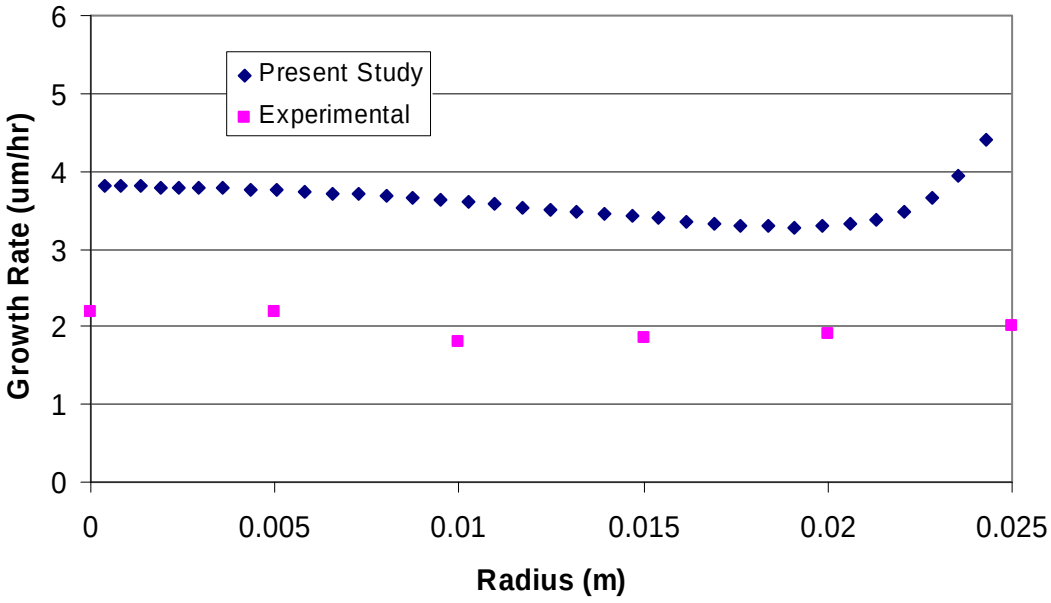

Figure 20: GaN Stagnation Reactor High Carrier Gas Flow Results [10]

These figures show that the growth rates of the simulation and the experimental are close in both cases. However, both figures show the numerical results having seemingly different trends than the experimental data. The first difference is that the

experimental data shows a drop in the growth rate between .0005 and .001 m (5cm and 1cm). The numerical data does not show an abrupt drop like the experimental data shows, nor does it show as great of a difference in growth rate between these two spots on the wafer. The numerical data does show the growth rate steadily dropping which seems physically reasonable. The reason for a smaller drop could be a difference in the circulation pattern inside the reactor, which is discussed in greater detail below. The other difference in the trends is that numerical results show a great increase in the growth rate on the outside edge of the wafer. Experimental results suggest only a slight increase, if any increase. This also could be due to a different circulation pattern in the reactor. This different circulation pattern could have increased amounts of recirculation of reacting particles causing more surface reactions causing a higher GaN growth rate. The different circulation pattern could be attributed to having different flow rates of the carrier gases through each inlet opening. The numerical approximation assumed that the carrier gas total flow rate was split evenly between the two inlets, and the experiment may have split it differently. Unfortunately, all necessary conditions under which the experiments were conducted are not available from the published reference for which the data was acquired, and therefore, only a qualitative comparison can be made.

3.1.3 Stagnation Point (Vertical) Reactor 2

A second vertical reactor was tested in which the III – V molar ratio was varied by changing the ammonia flow rate. The second reactor is the proprietary Thomas Swan reactor. The exact geometry of the reactor is unknown. Figure 21 shows a schematic sketch of the reactor. Some boundary conditions are available along with the

experimental data and these boundary conditions are shown below in Table 9. The other boundary conditions were assumed based on typical values from MOVPE reactors.

Boundary Condition	Value	Units
III/V Ratio	2200	n/a
Total Flow Rate	20	slm
Reactor Pressure	100	torr
Inactive Wall Temperature	333	ºK
Wafer Temperature	1303	ºK

Table 9: GaN Thomas Swan Operating Parameters [8]

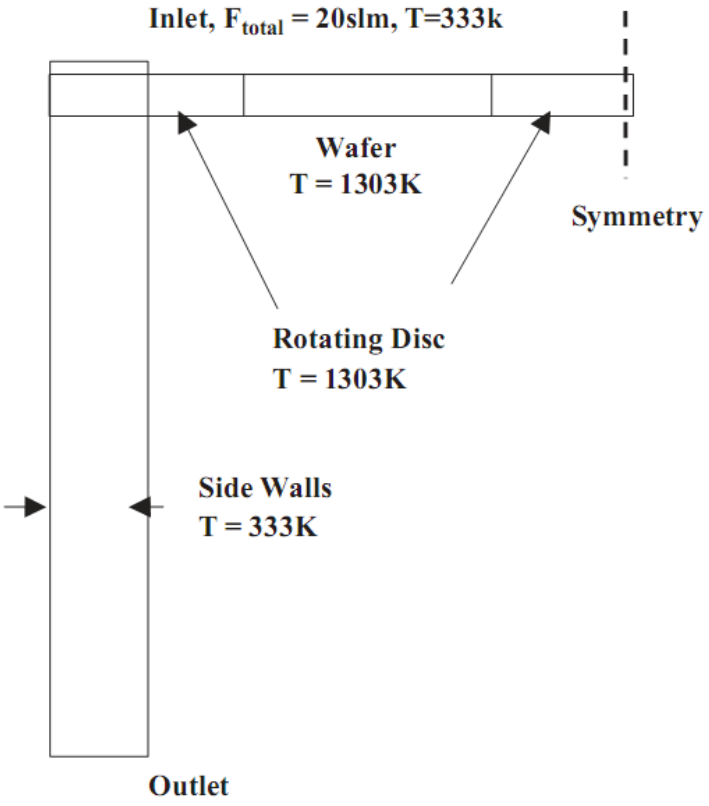

Figure 21: Thomas Swan Reactor [8]

Figure 22 shows a comparison of the experimental and the numerical model results for the Thomas Swan reactor.

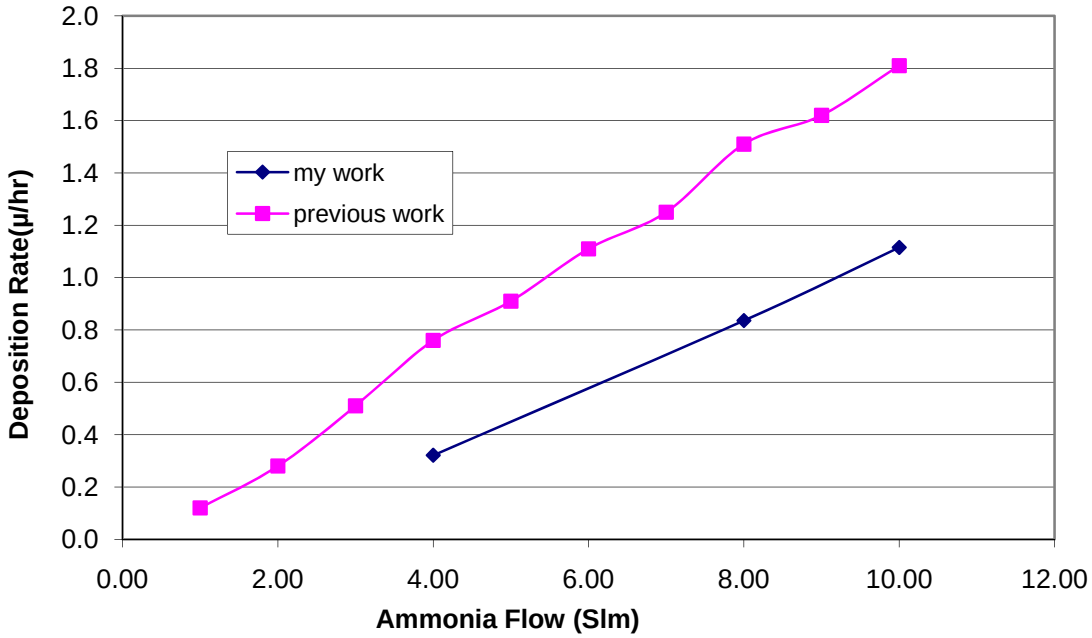

Figure 22: GaN Thomas Swan Results [8]

Both results show similar trends, highlighting the fact that numerical model captures all the relevant physics. The magnitudes differ because of the uncertainties in the geometry and operating conditions.

3.2 Steady State MOVPE of AlN

The preceding results lend credibility to the numerical model used in the study. In switching from GaN to AlN, the only aspect of the model that changes is the chemical reaction mechanism. Thus, a new study was conducted to test the validity of the reaction mechanism used for AlN growth.

3.2.1 Horizontal Reactor

Only one set of published experimental data is available to test the AlN model. This is an experiment in which the temperature is varied, much like what was done previously with the GaN model. The temperature dependence on the growth of AlN is

similar to that seen in the growth of GaN, but the regions occur at different values based on the different chemical reactions that dominate the growth of each material. The horizontal reactor geometry for AlN is the same reactor as the horizontal reactor for GaN. Figure 16 shows a sketch of the reactor. The boundary conditions are changed from the GaN model due to the different precursor gasses. The boundary conditions for the growth of AlN are shown in Table 10.

Boundary Condition	**Value**	**Units**
Reactor Pressure	85	torr
TMAL Flow Rate	10	μmol/min
H_2 Flow Rate	5000	sccm
NH_3 Flow Rate	1000	sccm
Inlet Temperature	300	ºK
Inactive Wall Temperature	300	ºK
Wafer Temperature	100 to 1500	ºC

Table 10: AlN Operating Condition for Steady State MOVPE of AlN [9]

Figure 23 shows a comparison of the experimental and the numerical model results.

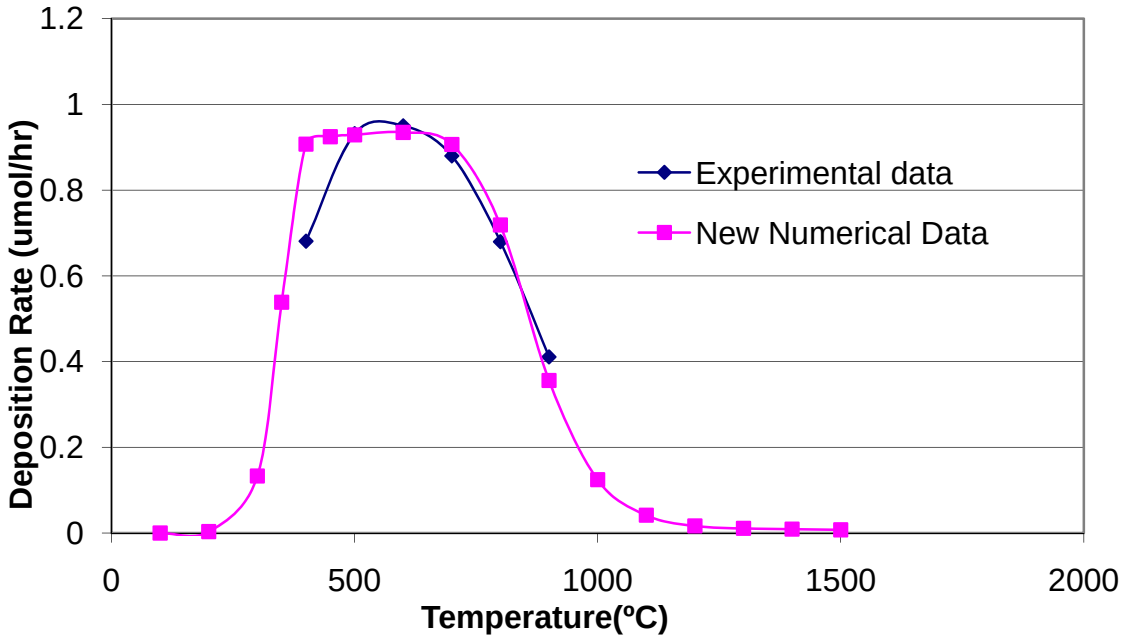

Figure 23: Comparison of Predicted Numerical Results against Previous Work for Steady State MOVPE of AlN at Various Temperatures [9]

The results agree fairly well. Additionally results look similar to the results seen for the steady state growth of GaN, with distinct regions showing kinetically limited growth, diffusion limited growth, and desorption-limited growth. A second parametric study was conducted in which the wafer surface temperature was held at constant temperature of 600ºC and the reactor pressure was varied between 30 torr and 270 torr, as summarized below as Table 11.

Boundary Condition	Value	Units
Reactor Pressure	30 to 270	torr
TMAL Flow Rate	10	μmol/min
H_2 Flow Rate	5000	sccm
NH_3 Flow Rate	1000	sccm
Inlet Temperature	300	ºK
Inactive Wall Temperature	300	ºK
Wafer Temperature	873	ºK

Table 11: AlN Horizontal Pressure Parameters [9]

Figure 24 shows the results of the numerical study compared to the experimental results.

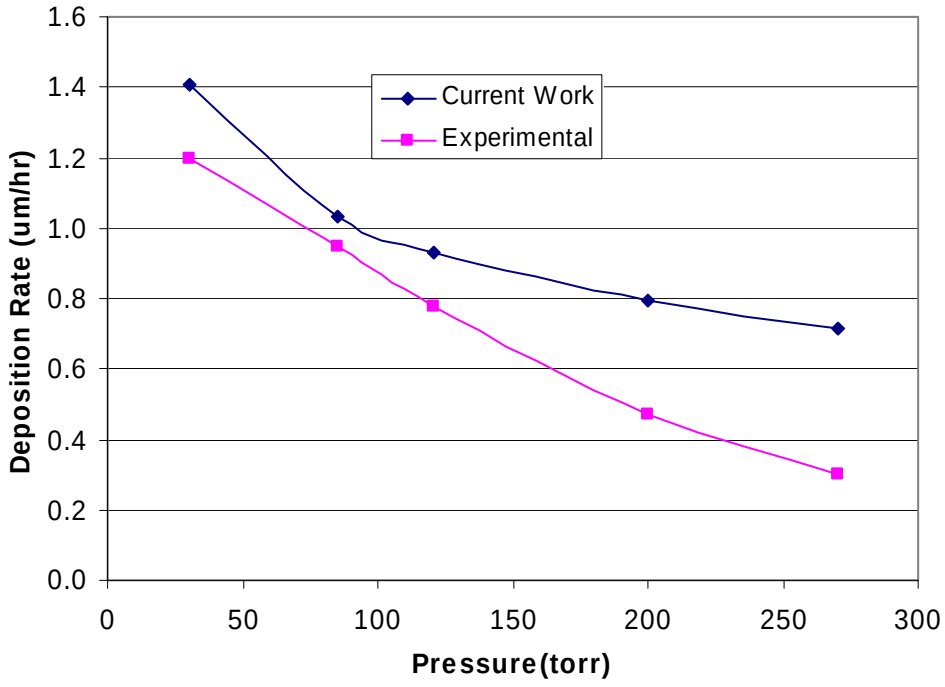

Figure 24: Effect of Pressure Variation on Steady State AlN MOVPE [9]

This shows that the model is accurate for various reactor pressures. Since the model is accurate when changing both the temperature and pressure, the reaction model was deemed appropriate for further studies of AlN growth.

3.2.2 Thomas Swan

An additional study was conducted for AlN growth in the Thomas Swan reactor to check the suitability of using the code for vertical reactors even though no experimental data is available for this reactor. Figure 21 shows a schematic of the Thomas Swan reactor. The boundary conditions that were used were the same as the GaN study, except replacing TMGA with TMAL. Only a single inlet flow rate was tested. The relevant boundary conditions are shown in Table 12.

Boundary Condition	Value	Units
III/V Ratio	2200	n/a
Total Flow Rate	20	slm
NH_3	4	slm
Reactor Pressure	100	torr
Inactive Wall Temperature	333	ºK
Wafer Temperature	873	ºK

Table 12: Operating Conditions for AlN MOVPE in the Thomas Swan Reactor

The numerical model predicted a growth rate of .7404 μm/hr of AlN, which is slightly higher than the predicted growth from the numerical modal for GaN. These results show that for this reactor the model is also suitable.

3.3 Pulsed MOVPE of AlN

The MOVPE of AlN results not only in the growth of an AlN thin film, but also in the growth of AlN particles suspended in the gas-phase. The growth of suspended

particles is undesirable due to the waste of incoming precursor gases, among other reasons. Pulsing of the precursor gases was proposed earlier as a way to prevent AlN particle formation, and is now tested numerically. The models used in the pulsed MOVPE are the same models used in the steady state tests, but are now exercised in unsteady mode with time-varying boundary conditions.

These models undergo a transient stage and then a periodic steady state, or a quasi steady state. During the transient stage the deposition (growth) rate and the AlN particle formation rate are changing. During the periodic steady state, the deposition rate and AlN particle formation rate are changing over time, but in a repeated pattern.

3.3.1 Horizontal Reactor

The first case studied is the simple horizontal reactor that was studied in great detail earlier. Figure 16 shows a sketch of the reactor. The operating parameters are shown in Table 13.

Boundary Condition	Value	Units
Reactor Pressure	85	torr
Nominal TMAL Flow Rate	10	μmol/min
H_2 Flow Rate	5000	sccm
Nominal NH_3 Flow Rate	1000	sccm
Inlet Temperature	300	ºK
Inactive Wall Temperature	300	ºK
Wafer Temperature	873	ºK

Table 13: AlN Transient Horizontal Reactor Properties

A nominal pulse time of .1 seconds and a time step of .01 seconds was used, implying that each pulse was split into 10 time-steps. This ensures that all events within

a pulse are adequately captured in the simulation. Tests were run to see what an appropriate pulse duration time is. The tests were of pulse durations (widths) of 1, .1, and .01 seconds. These were chosen because .1 seconds is approximately the time it takes a particle of fluid to travel across the wafer. In addition an order of magnitude below and above were tested to see if this would be appropriate. This test was done with no carrier gas purge between the precursor gases. The tests showed that the pulse duration does not significantly affect either the growth rate or the AlN particle formation, as shown in Table 14.

Pulse Width	AlN Deposition Rate	AlN Particle
sec	µm/hr	kg/s
1	1.23E-02	4.45E-03
0.1	1.29E-02	6.74E-03
0.01	1.19E-02	7.00E-03

Table 14: Pulse Width Testing Results

From a practical standpoint, a pulse width of 1 second incurs wastage of precursor gases, which would defeat one benefit of pulsing. The .01 second pulse width might be too small to implement practically. Therefore, a pulse width of .1 second is used for the remainder of the study. Figure 25 is a sample of the inlet flow rate for the normal sized pulse with a fully adjusted precursor gas flow rate. The flow rates are normalized so the pattern can be easily seen.

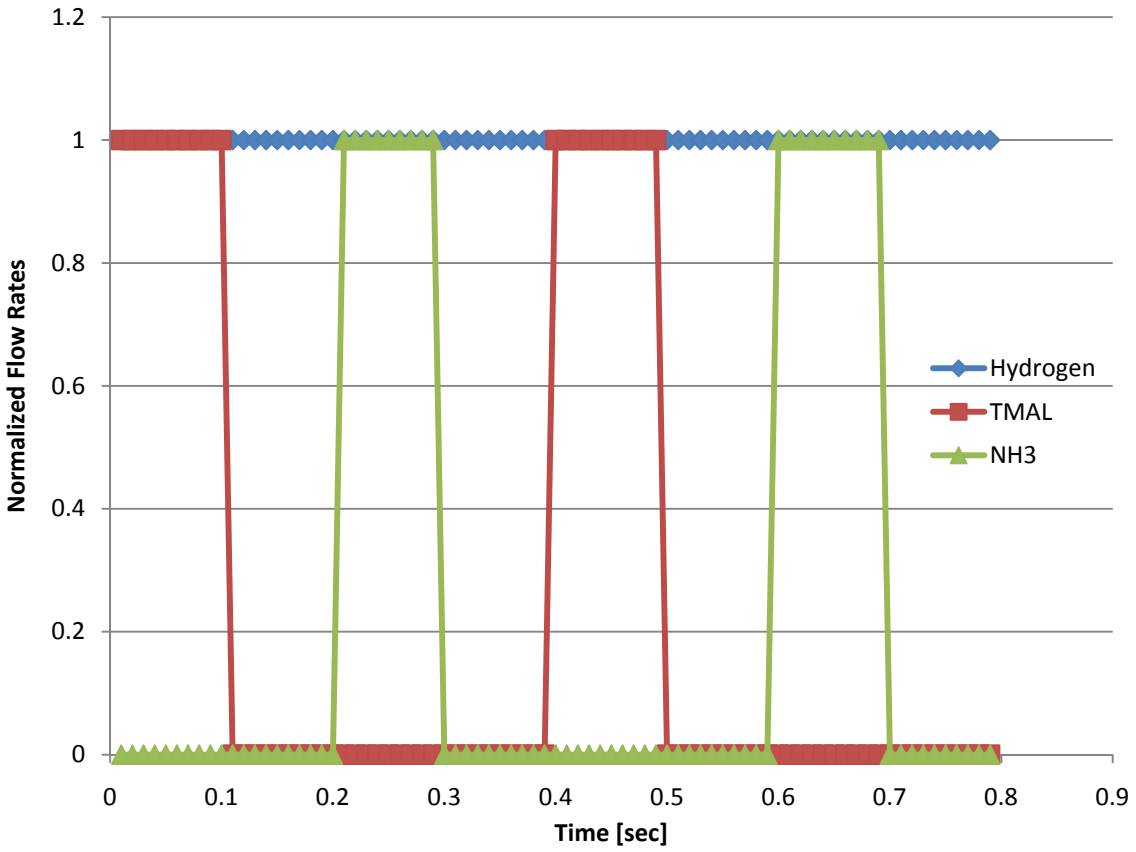

Figure 25: Sample Normalized Flow Rate Chart

All cases reached a quasi steady state after a transient stage. It reaches this state for all variables, but most notably for both the AlN film growth and AlN particle formation. The AlN particle formation reported is the total mass flow of AlN out of the exit of the reactor. Figure 26 shows an example of the transient stages and the quasi steady state for the deposition rate. Figure 27 shows an example of AlN particle formation over the transient and quasi steady state.

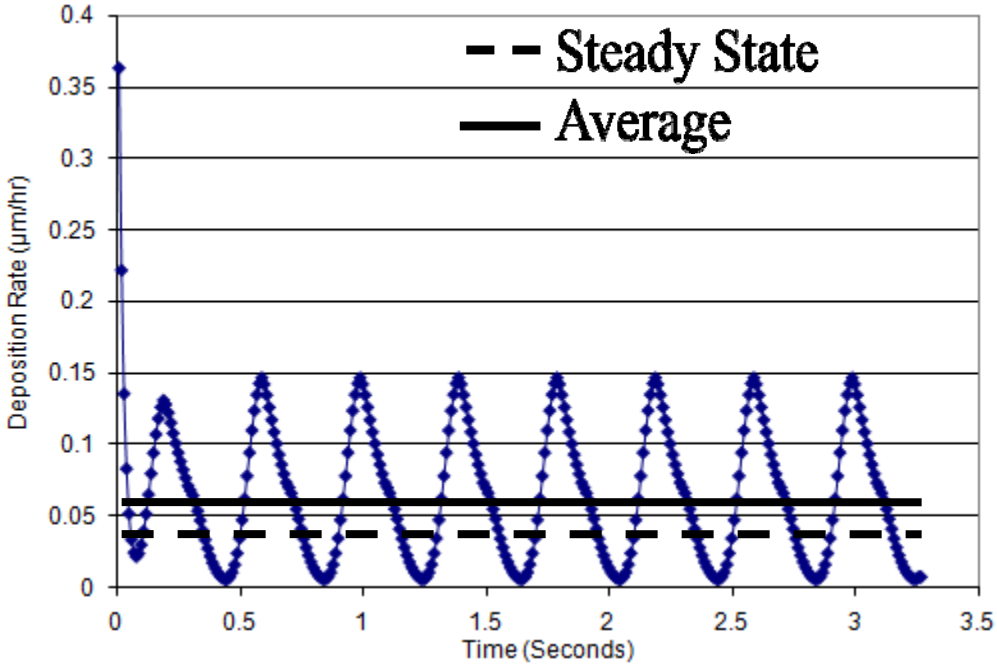

Figure 26: AlN Deposition Rate Transient Response

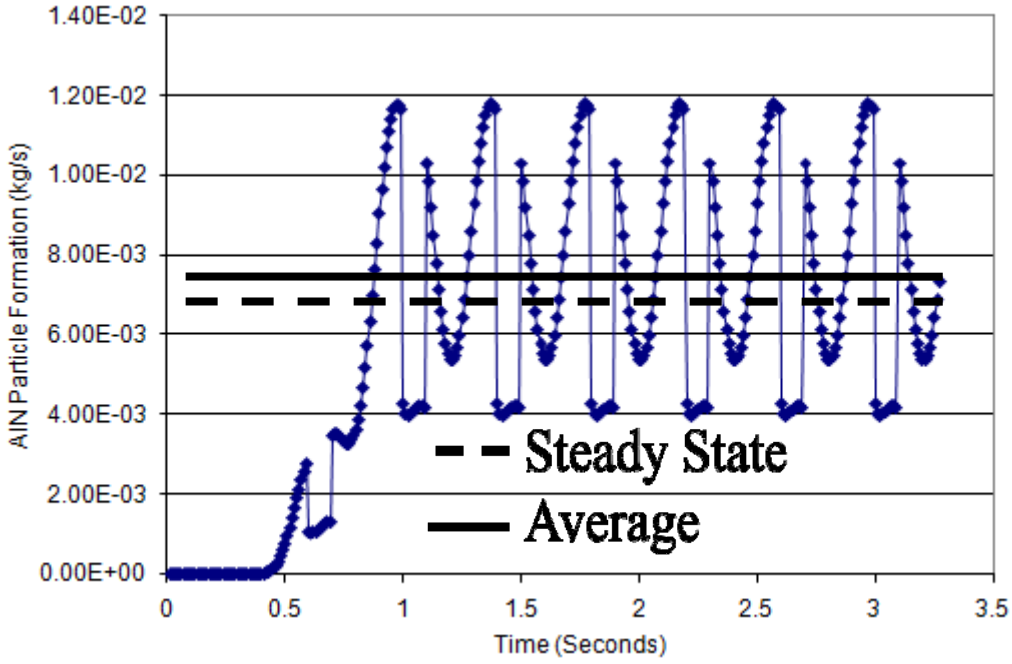

Figure 27: AlN Particle Formation Transient Response

From these figures it can be seen that it takes around 1 second, or 2 complete pulse cycles, before both AlN growth and AlN particle formation reach a quasi steady

state. That is, both AlN growth and AlN particle formation continue in a pattern that repeats itself in a periodic manner, referred to earlier as "quasi" steady state. The AlN deposition rate reaches quasi steady state before the AlN particle formation due to the AlN particle formation being calculated at the exit of the reactor. After the quasi steady state is reached, the pattern repeats indefinitely. The reported values for AlN deposition rate and AlN particle formation is the average over a pulse cycle once quasi steady state has been reached. This value is then compared with the growth rates obtained without pulsing (SS MOVPE) to gain an understanding of whether pulsed MOVPE is effective.

The pulse lengths for the carrier gas and the precursor gas flow rates were varied to see the effect on growth efficiency. Generally, in each carrier gas pulse length tested, the lowest precursor gas flow rate is the flow rate associated with using the original setup. This means that at this lowest gas flow rate if there was no pulsing, the gas flow rate would be the reported value in Table 13. However, since there is pulsing, there is less flow of the precursor gases over time. This means there is less precursor gas present in the reactor. Less gas means there are less reactant able to react and form into the desired AlN film. In order to counter this effect, and keep the same effective amount of precursor gas present in the reactor, the precursor gas flow rates are increased.

Quasi steady state responses were reached for each case presented below. The hydrogen pulse width was first varied and the results are shown below as Table 15. Figure 28 shows the results in graphical form. The hydrogen pulse width column refers to the length of the pulse time of the carrier gas. The flow rate of III/Vs column is the amount of precursor gas coming into the reactor, as compared to the original SS value coming in. The AlN particle formation and Deposition Rate are the amounts of AlN

solid particle and the AlN deposition rate respectively. The ALN/ALN (SS) column is the ratio of AlN particle formation in the pulsed case to the SS value, expressed as a percentage. The DEP/DEP (SS) is the same percentage as ALN/ALN (SS), but for the deposition rate. The AlN/TMAL column shows the amount of aluminum that leaves the reactor in the form of AlN particles compared to the amount of aluminum entering the reactor, and is a measure of how Al is wasted as particles.

Hydrogen Pulse Width	Percentage of Fully adjusted Flow Rate of III/Vs	AlN Particle Formation	AlN/AlN(SS)	Deposition Rate	DEP/ DEP(SS)	ALN/ TMAL
sec		kg/s		µm/sec		
Steady State		5.16E-03	N/A	3.88E-02	N/A	0.128%
0	50.00%	3.79E-03	73.46%	2.91E-02	74.9%	0.207%
0.1	25.00%	1.66E-03	32.19%	1.57E-02	40.5%	0.172%
0.2	16.67%	5.95E-04	11.53%	9.74E-03	25.1%	0.092%
0.3	12.50%	1.57E-04	3.03%	2.68E-03	6.9%	0.032%
0.5	8.33%	1.41E-07	0.00%	4.68E-03	12.1%	0.000%
0.7	6.25%	1.01E-09	0.00%	3.66E-03	9.4%	0.000%

Table 15: AlN Pulsing Results for Nominal Pulse Widths of .1 sec and Constant Precursor Gas Flow Rate in a Horizontal Reactor

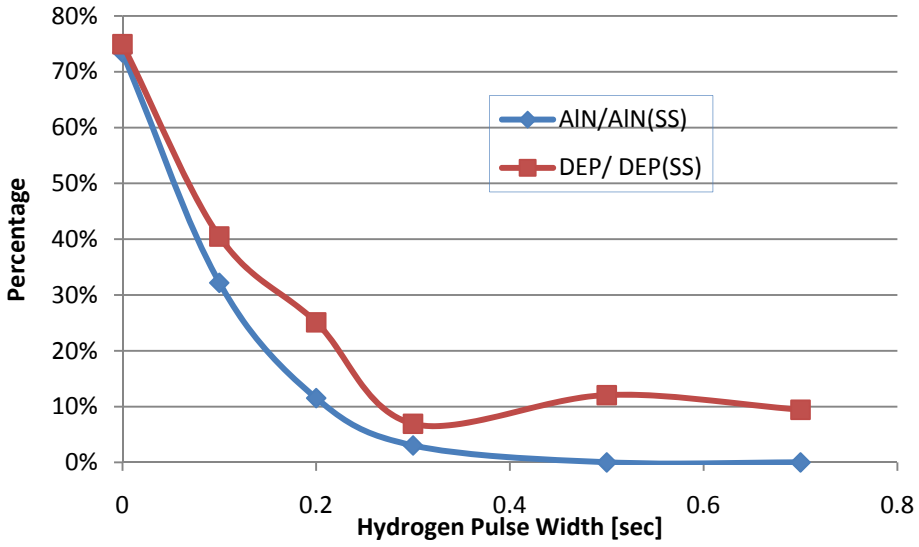

Figure 28: AlN Pulsing Results for Nominal Pulse Widths of .1 sec and Constant Precursor Gas Flow Rate in a Horizontal Reactor

These results show that both the AlN particle formation and AlN deposition rate drop as the hydrogen pulse length increases. These results could both be explained by the fact that there are less reactants present in the reactor. The tests were repeated with the precursor gas flow rate fully adjusted to see if the AlN particle formation is still slowed

down while maintaining deposition rates similar to the steady state values. A fully adjusted flow rate means that the average flow rate of the precursor gas over a pulse cycle is the same as the steady value. Table 16 shows the fully adjusted flow rate results. Figure 29 shows the fully adjusted flow rate results in graphical form.

Hydrogen Pulse Width	Percentage of Fully Adjusted Flow Rate of III/Vs	AlN Particle Formation	AlN/AlN(SS)	Deposition Rate	DEP/ DEP(SS)	ALN/ TMAL
sec		kg/s		µm/sec		
Steady State		5.16E-03	N/A	3.88E-02	N/A	0.128%
0.1	100%	7.33E-03	142.1%	6.51E-02	167.88%	0.182%
0.2	100%	3.44E-03	66.8%	6.00E-02	154.69%	0.085%
0.3	100%	6.11E-04	11.9%	5.72E-02	147.30%	0.015%
0.5	100%	1.13E-06	0.0%	5.56E-02	143.25%	0.000%
0.7	100%	1.03E-08	0.0%	5.40E-02	139.27%	0.000%

Table 16: AlN Pulsing Results for Nominal Pulse Widths of .1 sec and Fully Adjusted Precursor Gas Flow Rate in a Horizontal Reactor

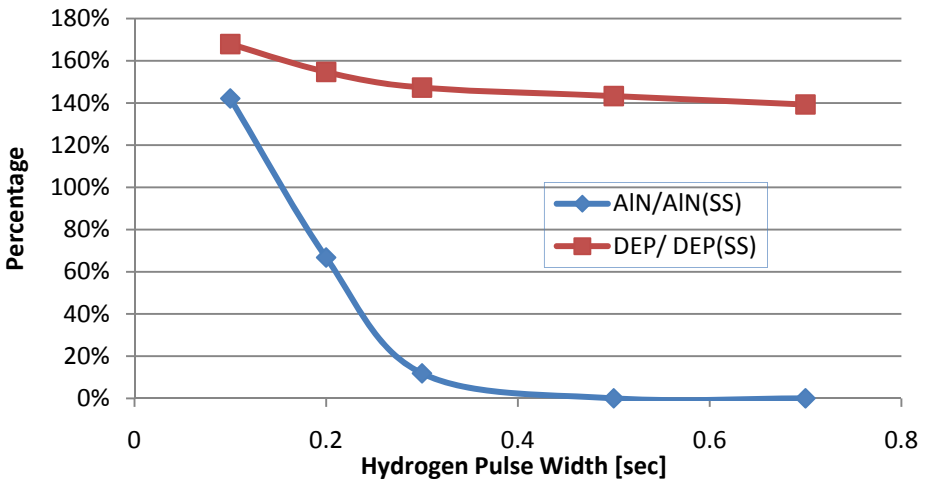

Figure 29: AlN Pulsing Results for Nominal Pulse Widths of .1 sec and Fully Adjusted Precursor Gas Flow Rate in a Horizontal Reactor

These results show that the AlN particle formation decreases drastically as the carrier gas pulse width is increased. The deposition rate also does decrease. However, it does not drop nearly as severely as the particle formation. This trend is further studied at

each carrier gas pulse width to see the results of varying the precursor gas flow rates. The results of the hydrogen pulse width of .2 seconds are presented below as Table 17.

Hydrogen Pulse Width	Percentage of Fully Adjusted Flow Rate of III/Vs	AlN Particle Formation	AlN/AlN(SS)	Deposition Rate	DEP/ DEP(SS)	ALN/ TMAL
sec		kg/s		µm/sec		
0.2	16.67%	5.95E-04	11.53%	9.74E-03	25.10%	0.092%
0.2	33.33%	1.24E-03	24.00%	2.03E-02	52.43%	0.092%
0.2	50.00%	1.83E-03	35.39%	3.04E-02	78.43%	0.091%
0.2	100.00%	3.44E-03	66.76%	6.00E-02	154.69%	0.085%

Table 17: AlN Pulsing Results for Hydrogen Pulse Widths of .2 sec and Varied Precursor Gas Flow Rate in a Horizontal Reactor

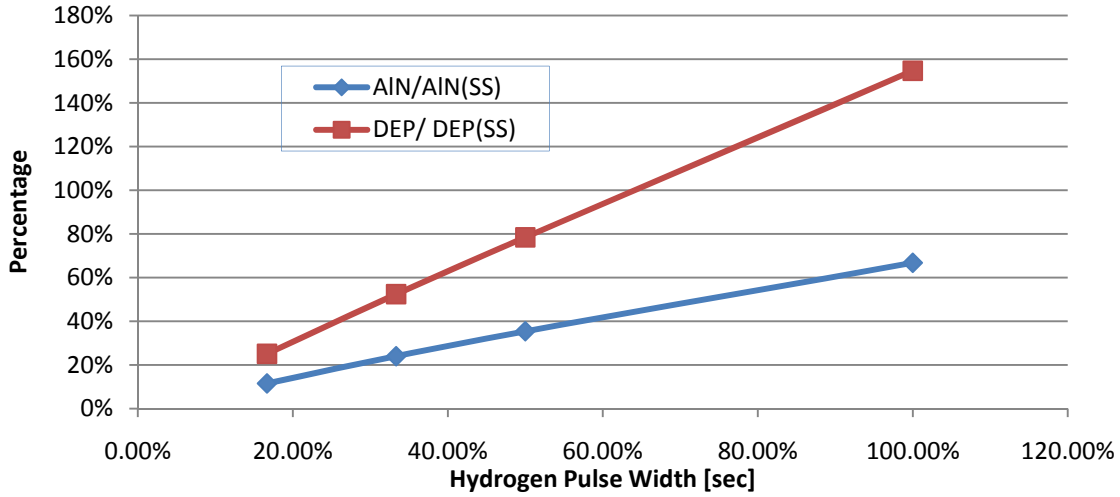

Figure 30: AlN Pulsing Results for Hydrogen Pulse Widths of .2 sec and Varied Precursor Gas Flow Rate in a Horizontal Reactor

Increasing the precursor gas flow rate increases both the deposition rate and AlN particle formation rate. This is expected because increasing the precursor gas flow rates increases the amount of reactant in the reactor, thus increasing the rate of reactions that can occur. The deposition rate increases at a greater rate than the particle formation rate and this is the result that is desired. Varying the precursor gas flow was repeated for all pulse width tests and this is shown in Table 18.

Hydrogen Pulse Width	Percentage of Fully adjusted Flow Rate of III/Vs	AlN Particle Formation	ALN/ ALN(SS)	Deposition Rate	DEP/ DEP(SS)	ALN/ TMAL
sec		kg/s		μm/sec		
Steady State		5.16E-03	N/A	3.88E-02	N/A	0.128%
0	0.5	3.79E-03	73.46%	2.91E-02	74.9%	0.207%
.1	0.25	1.66E-03	32.19%	1.57E-02	40.5%	0.171%
.1	0.5	3.60E-03	69.82%	3.29E-02	84.9%	0.178%
.1	1	7.33E-03	142.14%	6.51E-02	167.9%	0.181%
.2	0.1666	5.95E-04	11.53%	9.74E-03	25.1%	0.092%
.2	0.3333	1.24E-03	24.00%	2.03E-02	52.4%	0.092%
.2	0.5	1.83E-03	35.39%	3.04E-02	78.4%	0.090%
.2	1	3.44E-03	66.76%	6.00E-02	154.7%	0.085%
.3	0.125	1.57E-04	3.035%	2.68E-03	6.9%	0.032%
.3	0.5	3.63E-04	7.032%	2.91E-02	75.1%	0.017%
.3	1	6.11E-04	11.86%	5.72E-02	147.3%	0.015%
.5	0.08	1.41E-07	0.0027%	4.68E-03	12.1%	0.000%
.5	0.25	4.96E-07	0.010%	1.45E-02	37.5%	0.000%
.5	0.5	8.53E-07	0.017%	2.87E-02	74.0%	0.000%
.5	1	1.13E-06	0.022%	5.56E-02	143.2%	0.000%
.7	0.0625	1.01E-09	0.00002%	3.66E-03	9.4%	0.000%
.7	0.25	6.20E-09	0.00012%	1.45E-02	37.3%	0.000%
.7	0.5	1.01E-08	0.00020%	2.84E-02	73.3%	0.000%
.7	1	1.03E-08	0.00020%	5.40E-02	139.3%	0.000%

Table 18: Pulsed MOVPE AlN Results in a Horizontal Reactor

Each pulse width behaved as the .2 second pulse width did. Overall, there is a clear drop in AlN particle formation as the carrier gas purge length is increased, as well as a clear increase in AlN deposition rate as the flow rates of the precursor gases are increased.

3.3.2 Stagnation Point (Vertical) Reactor

Figure 21 shows the vertical Thomas Swan reactor used, the same reactor used previously. The operating conditions used are summarized in Table 19.

Boundary Condition	Value	Units
III/V Ratio	2200	n/a
Total Flow Rate	20	slm
Reactor Pressure	100	torr
Inactive Wall Temperature	333	K
Wafer Temperature	600	K

Table 19: Thomas Swan Transient AlN Reactor Properties

The Thomas Swan reactor geometry caused the fluid flow to be much more complex. The fluid flow included stagnation points and recirculation zones. The complexities of these cause the model to reach quasi steady state much slower.

Quasi steady state was reached for each of the cases presented below. Tests were run by varying pulse width and varying the precursor gas flow rates. The results are presented in Table 20.

Hydrogen Pulse Width	Percentage of Fully adjusted Flow Rate of III/Vs	AlN Particle Formation	AlN/AlN(SS)	Deposition Rate	DEP/ DEP(SS)	ALN/ TMAL
sec		kg/s		μm/sec		
Steady State		7.19E-03		1.23E-02		0.15%
0	50.00%	1.71E-03	23.74%	2.90E-03	23.00%	0.04%
0.1	25.00%	2.11E-03	29.33%	2.92E-03	24.00%	0.05%
0.1	100.00%	7.76E-03	107.87%	1.18E-02	96.00%	0.17%
0.5	8.30%	1.87E-03	25.94%	3.16E-03	26.00%	0.04%
0.5	100.00%	4.91E-03	68.32%	4.91E-02	398.10%	0.11%

Table 20: Transient AlN Thomas Swan Results

Compared to the results shown earlier for a horizontal reactor, these results do not follow a clear trend. However, the results at .5 second hydrogen pulse width show that there would be a benefit to pulsing the precursor and increasing the precursor gas flow rate. The increase in deposition rate over the steady state value could be explained by the

lower amount of aluminum wasted to form the suspended AlN particles and now the aluminum is available to grow on the wafer.

4 Summary and Conclusion

The idea of pulsing the precursors as a way to curb AlN particle formation was tested by numerical simulations, and was met with success. The numerical models show that AlN particle formation decreases significantly as the length of carrier gas purges increase and the deposition rate of substrate AlN can stay at approximately the same value as the SS value with the same precursor gas flow rates as used in SS cases. An added benefit is that since less precursor gases are wasted in the form of particles, these precursors actually contribute to the growth of the film. Thus the deposition rates (time averaged) obtained in the pulsed MOVPE cases are slightly higher than SS MOVPE deposition rates. To obtain these same growth rates requires more powerful pumps. If more powerful pumps are not used and pulsing is implemented, than there would be slower film growth rates. Slower growth rates require longer time required to grow the same thickness film and if the process is too slow than it may become impractical. Therefore, more powerful pumps would be needed to increase the precursor gas flow but these come at a greater cost to buy and operate. However, by using more powerful pumps and pulsing, AlN particle formation would go away almost entirely and if good deposition rates are maintained, as predicted, the whole process would be more cost efficient. There may be new problems that are not shown by the simulations, which come along with pulsing the precursor gas, such as different material properties for the grown substrate, or issues in the reactor and piping caused by the increased flow. The only way to find out if these problems could arise would be to run physical experiments and determine the results.

5 Future Work

There are more numerical tests that could be run. This particular research focused on the pattern of pulsing the group V precursor gas in, then carrier gas, followed by the group III precursor gas, and then the carrier gas again. There could be different patterns that could increase efficiency. Also, each reactor behaves differently, so it is important to run tests to determine what pulse length and pulsing pattern is most efficient for each reactor. For a specific reactor more pulse lengths and durations could be tested to hone in the most efficient process, but this must be done by balancing AlN film growth and all the associated costs associated with growth, including but not limited to precursor gas cost, pump costs, and maintenance costs.

The most important future work is that these numerical tests all need to be validated by actually performing physical experiment and measuring the growth rates and particle formation. Data from pulsing the precursor gases during the MOVPE of any III-V semiconductor would help verify that this process is worth investigating further.

6 References

1. Taniyasu, Y. et al. (2006). "An aluminum nitride light-emitting diode with a wavelength of 210 nanometers". Nature 441: 325.

2. http://www.icis.cnr.it/linee_ricerca/progettazione_molecolare/pm_prog5_rossetto/Movpe/aix/reactor.html Site accessed 1/14/2010 5:44

3. Stringfellow, G. B., (1999), *Organometallic Vapor-Phase Epitaxy: Theory and Practice*, Academic Press.

4. Creighton, J.R., Wang, G.T., and Coltrin, M.E., (2007), "Fundamental Chemistry and Modeling of Group-III Nitride MOVPE," *Journal of Crystal Growth*, Vol. 298, pp.2-7

5. Mihopoulos, T.G., Gupta, V., and Jensen, K.F., (1998), "A Reaction-Transport Model for AlGaN MOVPE Growth," *Journal of Crystal Growth*, Vol. 195, pp. 733-739

6. Krishnan, A. and Zhou, N.; "Analysis of Chemical Vapor Deposition in Industrial Reactors"; Fourth ASME/JSME Thermal Engineering Conference, Hawaii, March 19-24, 1995.

7. http://www.esi-cfd.com/content/view/80/192/ Site accessed 1/6/2010

8. Sengupta D., Mazumder S., Kuykendall W., and Lowry S. A., (2005), " Combined ab initio quantum chemistry and computation fluid dynamics calculations

of prediction of gallium nitride growth," *Journal of Crystal Growth*, Vol. 275, pp. 369-382

9. Chen C. H, Liu H., Steigerwald D., Imler W., Kuo C.P., Craford M.G., (1996), "A Study of Parasitic Reaction Between Nh_3 and TMGa and TMAl," *Journal of Electronic Materials*, Vol. 25, No.6, pp. 1004-1008

10. Safvi S.A., Redwing J.M., Tischler M.A., Kuech T.F., (1997), "GaN Growth by Metallorganic Vapor Phase Epitaxy," *Journal of The Electrochemical Society*, Vol. 144, No.5, pp. 1789-1796

Appendix A: Sample Code

```
!*********************************************************************
*
!
! Notes on how to declare and use user subroutines :
! Please Read this note before you start using the routines.
! For Declaring Global va riables :
!
! Declare all the global variables in cfdrc_user module. This will
! replace the use of common blocks (obsolete in FORTRAN 90).
!
! Include all the required global (common block) variables from the
! cfdrc_user module. For integer, real, and string length precisions
! use int_p, real_p, and string_length (recommended). For strings,
! string length can be less than string_length, but do not use more
! than string_length.

! Note: Include only required variables. This is a good way to protect
! the unused variables.

! Declare all your local variables inside your subroutines.
! For precisions use int_p and real_p for Integer and Real variables,
! respectively. Here real_p is synonymous with DOUBLE PRECISION.
! It is important to declare all the local variables (which are
! not included from cfdrc_user module.) used in this subroutine
! because IMPLICIT NONE is assumed. The compiler will give an error if
! some of the variables are not declared. Do not declare the variables
! which are included from the cfdrc_user module.

! Since Fortran 90 allows dynamic memory allocation, users are
! encouraged to take advantage. Variables and Array variables
! can be declared in the following manner: Multiple variables can be
! declared in a single line, with ',' as separator. Use '&' as the
! continuation character for multiple lines.

! Note again that string_length, real_p, and int_p have to be included
! from cfdrc_user module.
!
! for Real variables:
! REAL(real_p) :: variable names
! for one, two and three dimensional Real arrays:
! REAL(real_p), ALLOCATABLE, DIMENSION(:) :: variable names
! REAL(real_p), ALLOCATABLE, DIMENSION(:,:) :: variable names
! REAL(real_p), ALLOCATABLE, DIMENSION(:,:,:) :: variable names

! for Integer Variables:
! INTEGER(int_p) :: variable names
! for Integer arrays:
! INTEGER(int_p), ALLOCATABLE, DIMENSION(:) :: variable names

! for Logical Variables:
! LOGICAL :: variable names
! for Logical Arrays:
! LOGICAL, ALLOCATABLE, DIMENSION(:) :: variable names

! for Character Strings:
```

```
! CHARACTER(len=length of string) :: variable names
! for Character String arrays:
! CHARACTER(len=string_length), ALLOCATABLE, DIMENSION(:) :: variable
names

! Important:
! 1. Initialize all the local and global variables and arrays with
!    appropriate values. Initialize global variables in module only.
!
! 2. Use util parameters when necessary.
!
! 3. Try to write your variable declarations between
!   USER VARIABLE DECLARATION BEGIN and
!   USER VARIABLE DECLARATION END blocks in each subroutine.
!   Try to write your code declarations between
!   USER CODE BEGIN and USER CODE END blocks
!
! 4. All the names (Boundary, Volume, Zone) are case sensitive, except
!    variable names. Be careful when getting the indexes.
!
! 5. For all property routines, if users are using same formulation in
!    calculating the values (to be set), then user need not use the
!    condition for checking volume condition.
!
! 6. To get the total number of cells in this volume condition in
!    property related user subroutines use:
!
!    CALL get_cells_vc(vc_index, ncells, error)
!
!    then check the error message.
!    To get the corresponding global cell index, run the following
!    loop and get the global cell index.
!
!    DO ic = 1, ncells
!      CALL get_cell_index_from_vc(ic, vc_index, global_cell, error)
!      IF (error) THEN
!        call user_info(3, 'Error : in uvisc, after
get_cell_index_from_vc')
!        STOP
!      END IF
!      Before assigning to array allocate memory to array
!      then initialize to zeros.
!      vc_global_cell(ic) = global_cell
!    END DO
!
!    Using the global cell index, the user can get the value of
!    different variables at the cell center. Since all property
!    routines are called for each cell in the volume condition where
!    user defined variables are defined, the above procedure to get
!    the global cell indexes is not required unless user wants to
!    calculate average of any values at the beginning of the routine.
!
! 7. It is always advised not to pass character strings to subroutines.
!    Instead, assign character string to a variable, then pass
variable.
!
```

```
!**********************************************************************
*
!
!**********************************************************************
*
 MODULE cfdrc_user
!**********************************************************************
*
  IMPLICIT NONE
  SAVE

  INTEGER, PARAMETER :: int_p  = SELECTED_INT_KIND(8)

  INTEGER, PARAMETER :: string_length = 80

  INTEGER, PARAMETER :: real_p = SELECTED_REAL_KIND(8)

  ! DO NOT CHANGE THE PARAMETER VALUES. THESE ARE FOR USE ONLY.
  ! Direction parameters.
  INTEGER(int_p), PARAMETER :: XDIR = 1, YDIR = 2, ZDIR = 3

  ! geometry related parameters.
  INTEGER(int_p), PARAMETER :: GEOM_THREED = 1, GEOM_TWOD = 2, &
                               GEOM_TWOD_AXI = 3

  ! material related flags.
  INTEGER(int_p), PARAMETER :: MAT_GAS = 1, MAT_FLUID = 2, &
                               MAT_SOLID = 3, MAT_BLOCK = 4

  ! Grid Connectivity related parameters.
  INTEGER(int_p), PARAMETER :: TRI_CELL = 1, QUAD_CELL = 2, &
                               TET_CELL = 3, PYRAMID_CELL = 4, &
                               PRISM_CELL = 5, HEX_CELL = 6, &
                               POLY_CELL = 7

  INTEGER(int_p), PARAMETER :: LINE_FACE = 1, TRI_FACE = 2, &
                               QUAD_FACE = 3, POLY_FACE = 4

  ! Time option parameters.
  INTEGER(int_p), PARAMETER :: TIME_ORIGINAL = 0, TIME_PREVIOUS = 1, &
                               TIME_CURRENT = 2
  ! model parameters.
  INTEGER(int_p), PARAMETER :: MODEL_STEADY = 1, &
                               MODEL_STEADY_RESTART = 2, &
                               MODEL_TRANSIENT = 3, &
                               MODEL_TRANSIENT_RESTART = 4

  ! electric sub model options.
  INTEGER(int_p), PARAMETER :: ELECTRIC = 1, &
                               ELECTRIC_ELECTROSTATICS_FVM = 2, &
                               ELECTRIC_ELECTROSTATICS_BEM = 3, &
                               ELECTRIC_DC_CONDUCTION = 4, &
                               ELECTRIC_AC_CONDUCTION = 5, &
                               ELECTRIC_TIME_DOMAIN = 6

  ! global bc types.
  INTEGER(int_p), PARAMETER :: BC_TYPE_INLET = 1, &
```

```
                                BC_TYPE_WALL = 2, &
                                BC_TYPE_EXIT = 3, &
                                BC_TYPE_INTERFACE = 4, &
                                BC_TYPE_F_F_INTERFACE = 5, &
                                BC_TYPE_S_S_INTERFACE = 6, &
                                BC_TYPE_S_F_INTERFACE = 7, &
                                BC_TYPE_F_B_INTERFACE = 8, &
                                BC_TYPE_B_B_INTERFACE = 9, &
                                BC_TYPE_S_B_INTERFACE = 10, &
                                BC_TYPE_SYMM = 11, &
                                BC_TYPE_CYCLIC = 12, &
                                BC_TYPE_THINWALL = 13

! heat transfer bc subtypes
INTEGER(int_p), PARAMETER :: BC_HEAT_INOUT = 1, &
                             BC_HEAT_SYMM = 2, &
                             BC_HEAT_ISOTHERMAL = 3, &
                             BC_HEAT_ADIABATIC = 4, &
                             BC_HEAT_FIX_Q = 5, &
                             BC_HEAT_NEWTON = 6, &
                             BC_HEAT_EXT_RADIATION = 7, &
                             BC_HEAT_CONJUGATE_INTERFACE = 8, &
                             BC_HEAT_THINWALL = 9, &
                             BC_HEAT_CYCLIC = 10, &
                             BC_HEAT_INTERFACE = 11, &
                             BC_HEAT_MIXING_PLANE = 12, &
                             BC_HEAT_COUPLE = 13, &
                             BC_HEAT_CHIMERA = 14

! flow bc subtypes
INTEGER(int_p), PARAMETER :: BC_FLOW_FIXM_INLET = 1, &
                             BC_FLOW_FIXP_OUTLET = 2, &
                             BC_FLOW_WALL = 3, &
                             BC_FLOW_SYMM = 4, &
                             BC_FLOW_FIXP_EXTRAPOLAT_OUTLET = 5, &
                             BC_FLOW_FIXPT_INLET = 6, &
                             BC_FLOW_FIXP_INLET = 7, &
                             BC_FLOW_CYCLIC = 8, &
                             BC_FLOW_INTERFACE = 9, &
                             BC_FLOW_MIXING_PLANE = 10, &
                             BC_FLOW_COUPLE = 11, &
                             BC_FLOW_CHIMERA = 12


! electric bc subtypes.
INTEGER(int_p), PARAMETER :: BC_ELECTRIC_FIX_POTENTIAL = 1, &
                             BC_ELECTRIC_FIX_FLUX = 2, &
                             BC_ELECTRIC_SYMM = 3, &
                             BC_ELECTRIC_CYCLIC = 4, &
                             BC_ELECTRIC_DIEL_DIEL = 5, &
                             BC_ELECTRIC_FIX_CHARGE = 6, &
                             BC_ELECTRIC_IGNORE = 7, &
                             BC_ELECTRIC_ZERO_CURRENT = 8, &
                             BC_ELECTRIC_THIN_WALL = 9, &
                             BC_ELECTRIC_INTERFACE = 10

! semi bc subtypes.
```

```
  INTEGER(int_p), PARAMETER :: BC_SEMI_FIX_POTENTIAL = 1, &
                               BC_SEMI_FIX_CHARGE = 2, &
                               BC_SEMI_CONJUGATE_WALL = 3, &
                               BC_SEMI_CYCLIC = 4, &
                               BC_SEMI_INTERFACE = 5

  ! DTF I/O parameters.
  INTEGER(int_p), PARAMETER :: DTF_IO_VAR_LEN = 36, DTF_IO_UNIT_LEN =
36

  ! use the following parameter to get the cell/node data from DTF file
you
  ! want.
  INTEGER(int_p), PARAMETER :: USER_CURRENT_DTF_FILE = 1, &
                               USER_RESTART_DTF_FILE = 2

  ! DIRECTION parameters for istropic/anisotropic models.
  INTEGER(int_p), PARAMETER :: DIR_ISOTROPIC = 0, DIR_NORMAL = 1, &
                               DIR_TANGENTIAL_1 = 2, &
                               DIR_TANGENTIAL_2 = 3

  ! Utility parameters.
  REAL(real_p) , PARAMETER :: zero = 0.0d0, one = 1.0d0, two = 2.0d0, &
                              three = 3.d0, four = 4.0d0, &
                              pi = 3.1415926535898d0

! Declare global variables
! USER GLOBAL VARIABLE DECLARATION BEGIN

! USER GLOBAL VARIABLE DECLARATION END

 END MODULE cfdrc_user
!**********************************************************************
*
!
!**********************************************************************
*
 SUBROUTINE ubound(bc_index, var_index, face_index, xfc, yfc, zfc)
 !DEC$ ATTRIBUTES DLLEXPORT :: ubound
 !******* DO NOT REMOVE ABOVE LINE FOR MS WINDOWS OS ************
!**********************************************************************
*
!
! purpose : set boundary value of the current boundary variable.
!
! inputs  : bc_index Integer, global boundary index.
!           var_index Integer, global variable index.
!           face_index, global boundary face index.
!           xfc Real, x coordinate of boundary face center.
!           yfc Real, y coordinate of boundary face center.
!           zfc Real, z coordinate of boundary face center.
!
! This routine is called face by face basis for each bc record. Use
! get_var_index, get_bc_index and get_active_cell to get the variable
! index, boundary index and active cell index respectively.
!
! Use get_value_one_cell to get the values of different variables
```

```
! in a cell associated to face.
!
! Use set_bc() to set the value of current variable(var_index) at
! current face(face_index) along boundary(bc_index).
!-----------------------------------------------------------------------
-

! Include required global variables declared in cfdrc_user module.
  USE cfdrc_user, ONLY : int_p, real_p
  USE cfdrc_user_access

  IMPLICIT NONE

  ! Declaration of arguments of this subroutine.
  REAL(real_p),   INTENT(IN) :: xfc, yfc, zfc

  INTEGER(int_p), INTENT(IN) :: bc_index, var_index, face_index

! Declare required local variables here.
! USER VARIABLE DECLARATION BEGIN

  REAL(real_p) :: time, bc_value
  INTEGER(int_p) :: time_step_no, user_option, iteration
  INTEGER(int_p) ::  H2Var_index, TMALVar_index, NH3Var_index,
U_index,unit3
  LOGICAL :: er
  CHARACTER(len=50) :: user_string, H2, TMAL, NH3, u, mode3


! USER VARIABLE DECLARATION END

!-----------------------------------------------------------------------
-
! Start writing code here.
! USER CODE BEGIN

         CALL get_iteration(iteration, er)
         user_option =3

         H2= "H2"
         CALL get_var_index(H2, H2Var_index,er)
          TMAL= "TMAL"
         CALL get_var_index(TMAL,TMALVar_index,er)
          NH3= "NH3"
         CALL get_var_index(NH3,NH3Var_index,er)
         u = "U"
         CALL get_var_index(u, U_index,er)
         CALL get_time(time,time_step_no, er)

         IF (bc_index == 20)then
             IF (mod(time_step_no,420) <10) THEN

             IF(var_index == H2Var_index) THEN
                bc_value = 1
             ELSEIF(var_index == NH3Var_index) THEN
                bc_value = 0
             ELSEIF(var_index == TMALVar_index) THEN
```

```
            bc_value = 0
         ELSEIF(var_index == U_index) THEN
            bc_value = -.215348
         End IF

         ELSEIF (mod(time_step_no,420) < 210) THEN

         IF(var_index == H2Var_index) THEN
            bc_value = 1
         ELSEIF(var_index == NH3Var_index) THEN
            bc_value = 0
         ELSEIF(var_index == TMALVar_index) THEN
            bc_value = 0
         ELSEIF(var_index == U_index) THEN
            bc_value = -.215348
         End IF

         ELSEIF (mod(time_step_no,420) < 220) THEN

         IF(var_index == H2Var_index) THEN
            bc_value = .006953997
         ELSEIF(var_index == NH3Var_index) THEN
            bc_value = .993046003
         ELSEIF(var_index == TMALVar_index) THEN
            bc_value = 0
         ELSEIF(var_index == U_index) THEN
            bc_value = -3.833256

         End IF

         ELSEIF (mod(time_step_no,420 >= 220)) THEN

         IF(var_index == H2Var_index) THEN
            bc_value = 1
         ELSEIF(var_index == NH3Var_index) THEN
            bc_value = 0
         ELSEIF(var_index == TMALVar_index) THEN
            bc_value = 0
         ELSEIF(var_index == U_index) THEN
            bc_value = -.215348
         End IF
         End If

else
         IF (mod(time_step_no,420) <10) THEN

         IF(var_index == H2Var_index) THEN
            bc_value = .772400386
         ELSEIF(var_index == NH3Var_index) THEN
            bc_value = 0
         ELSEIF(var_index == TMALVar_index) THEN
            bc_value = .227599614
         ELSEIF(var_index == U_index) THEN
            bc_value = -1.135206
         End IF

         ELSEIF (mod(time_step_no,420) < 210) THEN
```

```
            IF(var_index == H2Var_index) THEN
               bc_value = 1
            ELSEIF(var_index == NH3Var_index) THEN
               bc_value = 0
            ELSEIF(var_index == TMALVar_index) THEN
               bc_value = 0
            ELSEIF(var_index == U_index) THEN
               bc_value = -1.1305787
            End IF

            ELSEIF (mod(time_step_no,420) < 220) THEN

            IF(var_index == H2Var_index) THEN
               bc_value = 1
            ELSEIF(var_index == NH3Var_index) THEN
               bc_value = 0
            ELSEIF(var_index == TMALVar_index) THEN
               bc_value = 0
            ELSEIF(var_index == U_index) THEN
               bc_value = -1.1305787

            End IF

            ELSEIF (mod(time_step_no,420) >= 220) THEN

            IF(var_index == H2Var_index) THEN
               bc_value = 1
            ELSEIF(var_index == NH3Var_index) THEN
               bc_value = 0
            ELSEIF(var_index == TMALVar_index) THEN
               bc_value = 0
            ELSEIF(var_index == U_index) THEN
               bc_value = -1.1305787
            End IF
            End If
         end if
      CALL set_bc(bc_value, er)

! USER CODE END
  RETURN
 END SUBROUTINE ubound


 SUBROUTINE uout(iflag)
 !DEC$ ATTRIBUTES DLLEXPORT :: uout
 !******* DO NOT REMOVE ABOVE LINE FOR MS WINDOWS OS ************
!*********************************************************************
*
!
! purpose : for customized user output.
!
! iflag:  flag indicating calling location.
!
! This routine is called 5 times at different instances of iterative
! cycle indicated by iflag.
!
```

```
! iflag  :
!          1 - At the beginning of RUN.
!              (At this point most of the boundary conditions,
!               properties are set. users should be able to get
!               cell or boundary values for different variables.
!               For non-moving grid problems, all of the grid
!               metrics are also available.)
!          2 - At the beginning of time step
!              (Only for transient problems. For moving grid,
!               grid metrics are available at this point.)
!          3 - At the end of each iteration.
!          4 - At the end of each time step.
!              (only for transient problems).
!          5 - At the end of RUN.
!
! One may use get_value_one_cell to get the values. To get the cell
! indexes, user has to supply the x,y,z locations and use the
! get_cell_index(vc_index,x,y,z,global_cell_index,error).
!-----------------------------------------------------------------------
-

! Include required global variables declared in cfdrc_user module.
  USE cfdrc_user, ONLY : int_p, real_p, string_length
  USE cfdrc_user_access

  IMPLICIT NONE

  INTEGER(int_p), INTENT(IN) :: iflag


  INTEGER(int_p) ::  DepVar_index, WaferBC_index, Wafer_faces,
Outlet_faces, unit, unit2
  INTEGER(int_p) :: WaferBC_type, WaferBC_face_index, user_option,
OutletBC_type
  INTEGER(int_p) ::  F_TMALVar_index, OutletBC_index,
F_AlNVar_index,InletBC_index
  INTEGER(int_p) ::  F_NH3Var_index, TMVar_index, HVar_index,
densVar_index,volVar_index, area_index
  LOGICAL :: er
  CHARACTER(len=50) :: model,model2, user_string, Dep, Wafer,
F_AlN,F_TMAL, Outlet, Inlet, F_NH3, totalMass, H, dens, vol,area

  REAL(real_p), DIMENSION(15):: bc_value
  REAL(real_p) :: bc_value2, F_AlNbc_value, F_TMALbc_value,
areabc_value , F_NH3bc_value, TMinBC_value, TMoutBC_value, Hbc_value

  INTEGER(int_p) :: vc_index, count, cell_index
  REAL(real_p) :: dens_value, vol_value, AlNmoles, TMALmoles, NH3moles,
Hmoles

! Declare required local variables here.
! USER VARIABLE DECLARATION BEGIN

! USER VARIABLE DECLARATION END

!-----------------------------------------------------------------------
-
```

```
! Start writing code here.
! USER CODE BEGIN

      unit = 4
      unit2 = 5
      model = "Dep.txt"
      model2 = "mole.txt"

      open (unit,file=model)
      open (unit2,file=model2)

      if(iflag== 1) THEN
      WRITE(unit, *) "First Dep Rate, Average Dep Rate,"
      WRITE(unit2, *) "TMAL in,AlN out,NH3 int"

      END IF

      IF(iflag>= 4) THEN

      Dep= "TOTAL_DEP_THICKNES"
      CALL get_var_index(Dep, DepVar_index,er)
      Wafer ="Wafer"
      CALL get_bc_index(Wafer, WaferBC_index,er)
      CALL get_bc_info(WaferBC_index, WaferBC_type,Wafer_faces, er)
      CALL get_bc(WaferBC_index,DepVar_index,Wafer_faces, bc_value, er)

      CALL
get_avg_value_one_bc_patch(WaferBC_index,DepVar_index,bc_value2,er)

      F_AlN = "FLUX_AlN_P"
      CALL get_var_index(F_AlN, F_AlNVar_index,er)
      Outlet = "Outlet"
      CALL get_bc_index(Outlet, OutletBC_index,er)
      CALL
get_avg_value_one_bc_patch(OutletBC_index,F_AlNVar_index,F_AlNbc_value,
er)

      F_TMAL = "FLUX_TMAL"
      CALL get_var_index(F_TMAL, F_TMALVar_index,er)

      Inlet = "TMAL"
      CALL get_bc_index(Inlet, InletBC_index,er)
      CALL
get_avg_value_one_bc_patch(InletBC_index,F_TMALVar_index,F_TMALbc_value
,er)

      Inlet = "NH3"
      F_NH3 = "FLUX_NH3"
      CALL get_var_index(F_NH3, F_NH3Var_index,er)
      CALL
get_avg_value_one_bc_patch(InletBC_index,F_NH3Var_index,F_NH3bc_value,e
r)

       H ="FLUX_H2"
      CALL get_var_index(H, HVar_index,er)
      CALL
get_avg_value_one_bc_patch(InletBC_index,HVar_index,Hbc_value,er)
```

```
      area = "Area"
      CALL get_var_index(area, area_index,er)
      CALL
get_avg_value_one_bc_patch(OutletBC_index,area_index,areaBC_value,er)

      CALL get_bc_info(OutletBC_index, OutletBC_type,Outlet_faces, er)

      TMALmoles = F_TMALbc_value*
areaBC_value*Outlet_faces/(144*.001/60*.000001)
      NH3moles = F_NH3bc_value*
areaBC_value*Outlet_faces/(17*.001/60*.000001)
      AlNmoles =    F_AlNbc_value*
areaBC_value*Outlet_faces/(41*.001/60*.000001)
      Hmoles = Hbc_value* areaBC_value*Outlet_faces/(2*.001/60*.000001)

      if(iflag == 5) THEN
      WRITE(unit, *) "final"
      WRITE(unit2, *) "final"
      end if

      WRITE(unit, *) -bc_value(1)*60000000,"," ,-
(bc_value2*60000000),",", Hmoles

      WRITE(unit2, *) TMALmoles,",",-AlNmoles,",", NH3moles
      end if
! USER CODE END
  RETURN
 END SUBROUTINE uout
!***********************************************************************
*
!
!***********************************************************************
*
```